

"Made available under NASA sponsorship  
in the interest of early and wide dis-  
semination of Earth Resources Survey  
Program information and without liability  
for any use made thereof."

"AS-BUILT" DESIGN SPECIFICATION  
FOR THE  
YIELD ESTIMATION SUBSYSTEM (YES)  
OPERATIONAL ROBERTSON PHENOLOGICAL  
MODEL

JSC-12941  
**80-10170**  
NASA CR-  
160680

Job Order 74-963  
AD 63-1347-4963-11

(E80-10170) AS-BUILT DESIGN SPECIFICATION  
FOR THE YIELD ESTIMATION SUBSYSTEM (YES)  
OPERATIONAL ROBERTSON PHENOLOGICAL MODEL  
(Lockheed Electronics Co.) 118 p  
HC A06/MF A01

N80-28771

Unclas  
00170  
CSCL 02C G3/43

Prepared By  
Lockheed Electronics Company, Inc.  
System and Services Division  
Houston, Texas  
Contract NAS 9-15200  
For  
EARTH OBSERVATIONS DIVISION  
SPACE AND LIFE SCIENCES DIRECTORATE



*National Aeronautics and Space Administration*  
**LYNDON B. JOHNSON SPACE CENTER**

*Houston, Texas*

June 1977

LEC-10743



JSC-12941

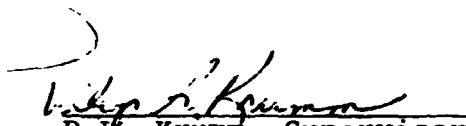
"AS-BUILT" DESIGN SPECIFICATION  
FOR THE  
YIELD ESTIMATION SUBSYSTEM (YES)  
OPERATIONAL ROBERTSON PHENOLOGICAL  
MODEL

Job Order 74-963  
AD 63-1347-4963-11

PREPARED BY

K. Williams

APPROVED BY

  
P.W. Krumm, Supervisor  
Software Development Section

Prepared By

Lockheed Electronics Company, Inc.

For

Earth Observations Division

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION  
LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

June 1977

## CONTENTS

Section	Page
1. SCOPE . . . . .	1-1
2. APPLICABLE DOCUMENTATION . . . . .	2-1
3. SYSTEM DESCRIPTION . . . . .	3-1
3.1 <u>HARDWARE DESCRIPTION</u> . . . . .	3-2
3.2 <u>SOFTWARE DESCRIPTION</u> . . . . .	3-2
3.2.1 TAPE AND DISK DATASETS AND ATTRIBUTES . . .	3-7
3.2.2 SUBROUTINE INIT . . . . .	3-10
3.2.3 SUBROUTINE SWAP . . . . .	3-13
3.2.4 SUBROUTINE TERM . . . . .	3-20
3.2.5 PROGRAM CLEAR . . . . .	3-23
3.2.6 JCL FILE CLEAR . . . . .	3-26
3.2.7 PROGRAM CAPTURE . . . . .	3-27
3.2.8 JCL FILE CAPTURE . . . . .	3-34
3.2.9 SUBROUTINE CVB . . . . .	3-35
3.2.10 PROGRAM PRINT . . . . .	3-38
3.2.11 JCL FILE LIST . . . . .	3-41
3.2.12 PROGRAM EDITOR . . . . .	3-42
3.2.13 JCL FILE EDIT . . . . .	3-48
3.2.14 SUBROUTINE CHECK . . . . .	3-49
3.2.15 SUBROUTINE REPLAC . . . . .	3-59
3.2.16 PROGRAM CROPCALN . . . . .	3-67
3.2.17 JCL FILE SUZYQ . . . . .	3-74

Section	Page
3.2.18 SUBROUTINE START . . . . .	3-76
3.2.19 SUBROUTINE PHENO . . . . .	3-79
3.2.20 FUNCTION DL . . . . .	3-84
3.2.21 PROGRAM INSDEL . . . . .	3-87
3.2.22 JCL FILE INSERT . . . . .	3-93
3.2.23 PROGRAM DRECOVER . . . . .	3-94
3.2.24 PROGRAM JSCTAPE . . . . .	3-97
4. OPERATION	
4.1 <u>USER DOCUMENTATION</u> . . . . .	4-1
4.2 <u>PROCEDURES</u> . . . . .	4-1
4.2.1 CONSTRUCTING THE INDEX FILE (DRECOVER) . . .	4-2
4.2.2 CROP CALENDAR PROCEDURES . . . . .	4-2
4.3 <u>FLOWCHART</u> . . . . .	4-12



## 1. SCOPE

This document describes the Yield Estimation Subsystem (YES) operational implementation of Robertson's Phenological Model for small grains. Specifically, this document describes the software established to compute Robertson's biometeorological time scale for all LACIE weather stations from approximations to daily maximum and minimum temperatures. This document does not describe the software at NMC that extracts the weather data or the software at JSC that writes the AI crop calendar reports.

~~SECRET~~

## 2. APPLICABLE DOCUMENTATION

- 1) Operating Procedures for the LACIE/YES Gridded Crop Calendar Report Writer by D.D. Wilcox. LEC-9955, Dec., 1976.
- 2) System Description of the LACIE/YES Gridded Crop Calendar Report Writer by D.D. Wilcox, LEC-10004, Jan., 1977.
- 3) AD 63-1347-4963-11 Documentation of Automated Crop Calendar Operation.
- 4) "A Biometeorological Time Scale for a Cereal Crop Involving Day and Night Temperatures and Photoperiod". G.W. Robertson, Intl. J. Biomet., 12#3 (191-223), (1968).
- 5) Level 3 Baseline  
Yield Estimation Subsystem (YES)  
Requirements  
LACIE C00200 Sept., 1976
- 6) Level 3 Baseline  
NOAA/NASA/USDA  
Interface Control Document  
LACIE C00710 Sept., 1976
- 7) A FORTRAN Implementation of the Robertson Phenological Model, D.D. Wilcox, G. Champagne, R. Baskett, S. Wooley. LEC Tech. Memo, LEC-5974.

### 3. SYSTEM DESCRIPTION

The Robertson Phenological Model is an algorithm for converting point estimates of daily maximum and minimum temperature and daylength to a point estimate of Robertson's Bio-meteorological Time Scale, a numerical scale describing the development of wheat. Knowledge of the stage of wheat development is useful to analyst interpreters (AI) in the estimation of wheat acreages. The Robertson Model has been implemented to make this knowledge available.

The Robertson Scale ranges continuously from 1.0, called planting, to 6.0, called ripe. The Robertson Model assigns to each integral range 1.0 to 2.0, ..., 5.0 to 6.0, an equation  $f_{1.0-2.0}(t_{\max}, t_{\min}, \text{daylength}), \dots, f_{5.0-6.0}(t_{\max}, t_{\min}, \text{daylength})$ . To advance the model one day, let  $D_{\text{yesterday}}$  be the development yesterday. From  $D_{\text{yesterday}}$  select one of  $f_{1.0-2.0}, \dots, f_{5.0-6.0}$  as  $f_{\text{yesterday}}$ . Obtain yesterday's  $t_{\max}, t_{\min}$ , and daylength, and compute:

$$D_{\text{today}} = D_{\text{yesterday}} + f_{\text{yesterday}}(t_{\max}, t_{\min}, \text{daylength}).$$

Clearly  $t_{\max}$  and  $t_{\min}$  are observations at the weather stations. It is most useful to have  $D_{\text{today}}$  at the sample segments (LACIE modelling region, for example). Therefore, some method of extending the weather station developments to the sample segments is necessary.

It should now be unequivocal that any implementation of the Robertson Model must have three major components:

- 1) A weather data collecting component,
- 2) a crop modelling component, and
- 3) a component that extends the output of (2) to the sample segments and writes a report.

This document describes the structure and operation of component 2. Component 3 is described in documents 1 and 2 of section 2.0 of this report. Component 1 is in the implementation stage. However, some software is described here which performs data collection and editing. It is anticipated these functions of component 2 will be abandoned when the final version of component 1 is put to use. From this point, component 2 will be referred to as 'the system'.

### 3.1 HARDWARE DESCRIPTION

These programs and datasets are resident on the IBM 360/195 complex at Suitland, Maryland. They should be transferable to any IBM 360-370 series machine with sufficient disk to handle the datasets and main memory to support the PL/I optimizing compiler.

### 3.2 SOFTWARE DESCRIPTION

The software consists of eight programs: one to prepare the station index file, six to prepare the crop calendar output, and one to copy this output from disk to tape. The program to prepare the station index file, DRECOVER, is input by cards at CCEA/Columbia. The six crop calendar preparation source programs are kept on cards at CCEA/Columbia and are listed in executable form on the partitioned dataset W.EDS. CCEA.PHASE3.LOAD; the executable programs are called by the following JCL files maintained on card decks at CCEA/Columbia:

- 1) CLEAR, used to initialize disk file CROPDATA,
- 2) CAPTURE, used to build the file,
- 3) LIST, which lists CROPDATA and can be accessed at any time during preparation of the file.
- 4) EDIT, which edits the file for mistakes and missing data,
- 5) SUZYQ, which performs crop modelling, and
- 6) INSERT, used to enter corrections into CROPDATA.

JSCTAPE, the program copying the crop calendar output to tape, is input by cards at CCEA/Columbia each time it is executed.

In general, a crop calendar update will require CLEAR, followed by several runs of CAPTURE, followed by LIST. At this point, EDIT is run to correct for missing data and mistakes. SUZYQ is run producing updates; any errors not corrected by EDIT are detected by SUZYQ. INSERT is then used to complete editing of the weather dataset and SUZYQ is rerun. After the final run of SUZYQ, JSCTAPE is run to produce a tape for transmittal to JSC.

Two (logical) files are maintained by SUZYQ to record the last predicted crop stage at each update, these files being in an Old Master/New Master relationship. In this way an error in

the preparation of the data (ex., incorrect data captured, not enough data captured) can be handled by going back to the Old Master and beginning the process anew. See Section 3.2.1.6 for further discussion. Data flow and file usage are shown in Figure 3-1.

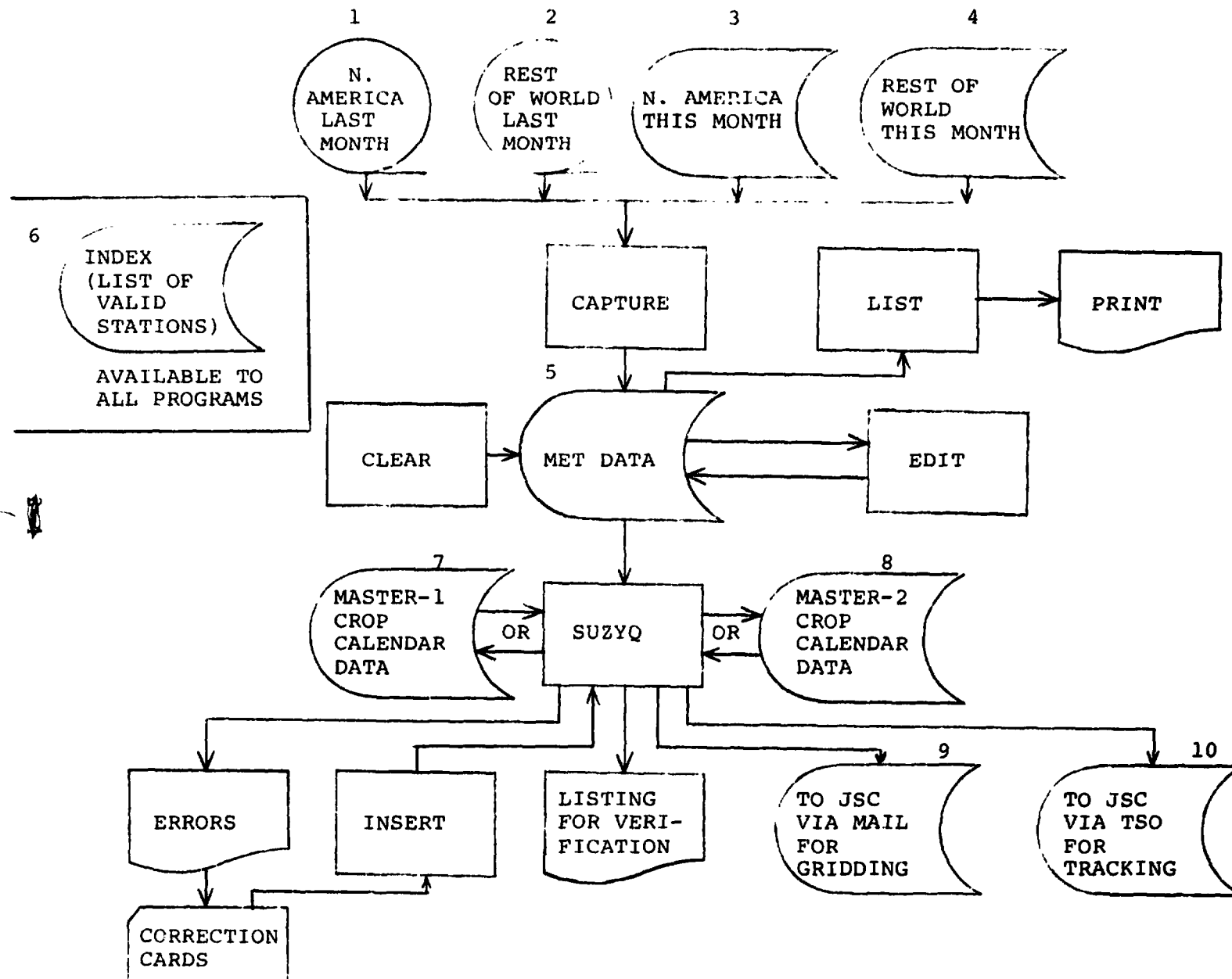


Figure 3-1.— Data Flow and File Usage in Crop Calendar System

Figure 3-1 (continued)

DATASET NAMES AND FLOWCHART REFERENCES

Flowchart Box #	DSN
1	CCEA.xxx.SRTD (see Section 3.2.1.1)
2	CCEA.DAILY.xxx (see 3.2.1.2)
3	W.EDS.CCEA.DAILY (see 3.2.1.3)
4	W.EDS.CCEA.DAYDATA (see 3.2.1.4)
5	W.EDS.CCEA.GROUPi.CROPDATA (see 3.2.1.7)
6	W.EDS.CCEA.GROUPi.INDEX (see 3.2.1.5)
7	W.EDS.CCEA.GROUPi.MASTER1 (see 3.2.1.6)
8	W.EDS.CCEA.GROUPi.MASTER2 (see 3.2.1.6)
9	W.EDS.CCEA.GROUPi.GRIDINFO (see 3.2.1.8)
10	W.EDS.CCEA.GRCUPi.ARNO (see 3.2.1.9)

xxx = month of interest, 3 letter abbreviation; ex. JAN,FEB,MAR

i = I for Group I countries(U.S.A., INDIA, CHINA)  
= II for Group II countries(U.S.S.R., CANADA)



The software components are divided into four classes:

- 1) Weather Data Access Routines:  
These are a set of routines, INIT, SWAP, and TERM which provide a uniform method of reading and writing data to the weather data files.
- 2) File Manipulation Routines:  
These routines include the main programs CLEAR, which initializes the weather data file, CAPTURE, which builds it, EDITOR, which edits out most mistakes and missing values, INSDEL, which makes card-input changes in the file, and PRINT which lists it.

Also included is the program DRECOVER which sets up the INDEX file, the supporting routine CVB located in the PDS W.EDS.CCEA.PROD.LOAD and called by CAPTURE, and the JCL files which are needed for invocation of the main programs.

The following is a list of the JCL files and their corresponding main programs:

<u>File Name</u>	<u>Program</u>
CLEAR	CLEAR
CAPTURE	CAPTURE
EDIT	EDITOR
INSERT	INSDEL
LIST	PRINT

- 3) Crop Modelling Routine:  
This includes the main program CROPCALN, supporting routines START and PHENO, function DL, and the JCL file SUZYQ, necessary for invocation.
- 4) Output Tape to JSC:  
This is a deck of JCL and the program JSCTAPE that produces the interface tape for shipment to JSC.



### 3.2.1 TAPE AND DISK DATASETS AND ATTRIBUTES

The following datasets will be referred to in subsequent descriptions of programs and subroutines.

#### 3.2.1.1 CCEA.xxx.SRTD

North American data for past months on tape, where 'xxx' is the month abbreviation (ex., Jan. Feb., etc.)

RECFM = FB	UNIT = TAPE 9
LRECL = 560	VOL = Eyyyyy, where 'yyyyy'
BLKSIZE = 5600	is the tape number

This dataset is accessed through the CAPTURE routine; a separate tape exists at Suitland for each month of past data and is differentiated from the others by both the DSN and the tape number.

#### 3.2.1.2 CCEA.DAILY.xxx

Non-North American data for past months on tape, where 'xxx' is the month abbreviation.

LRECL = 384  
BLKSIZE = 12672

All other information is identical to CCEA.xxx.SRTD discussed in 3.2.1.1.

#### 3.2.1.3 W.EDS.CCEA.DAILY

North American temperature data for the current month on disk.

RECFM = FB  
LRECL = 560  
BLKSIZE = 5600

This dataset is accessed during CAPTURE.

#### 3.2.1.4 W.EDS.CCEA.DAYDATA

Non-North American temperature data for the current month on disk.

CRECL = 384  
BLKSIZE = 12672



### 3.2.1.7 W.EDS.CCEA. $\left\{ \begin{array}{l} \text{GROUP I} \\ \text{GROUP II} \end{array} \right\}$ .CROPDATA

RECFM = F  
LRECL = 1024  
BLKSIZE = 1024  
DSORG = PS

This dataset contains the maximum and minimum daily air temperatures (or approximations thereto) for the 14 days for each station listed in INDEX. The CLEAR program is executed to change the starting day of the 14-day period, and also to initialize all values to '9999'. The CAPTURE program fills CROPDATA, and SUZYQ utilizes the file.

### 3.2.1.8 W.EDS.CCEA. $\left\{ \begin{array}{l} \text{GROUP I} \\ \text{GROUP II} \end{array} \right\}$ .GRIDINFO

RECFM = FB  
LRECL = 80  
BLKSIZE = 12960  
DSORG = PS

The GRIDINFO dataset is created or modified as output during execution of SUZYQ. The record contents are in the same format as those of the output master file; while the output master file contains information on only the last day of a two-week period, GRIDINFO contains similar information for every day in the two-week period. GRIDINFO contains only information on stations with successful crop calendar executions; as a result, it is necessary to complete an error-free run of SUZYQ before this dataset is copied to tape using program JSCTAPE.

### 3.2.1.9 W.EDS.CCEA. $\left\{ \begin{array}{l} \text{GROUP I} \\ \text{GROUP II} \end{array} \right\}$ .ARNO

RECFM = VBS  
LRECL = 796  
BLKSIZE = 800  
DSORG = PS

This dataset is created by an IEBPTPCH step after SUZYQ is successfully run, and contains basically the same information as the output master file. It is accessed by JSC via a dial-up telephone line using the IBM Time Sharing Operation (TSO).

### 3.2.2 SUBROUTINE INIT

INIT is called by CLEAR, CAPTURE, PRINT, EDITOR, INSDEL, and CROPCALN; INIT prepares the weather data file, CROPDATA, for direct access and initializes several variables used in SWAP.

#### 3.2.2.1 Linkages

INIT defines COMMON blocks TALK, INDEX, DATA, and CLOSE, which are used by other subroutines and programs.

#### 3.2.2.2 Interfaces

INIT must be called before SWAP.

#### 3.2.2.3 Inputs

INIT assumes that Unit 2 is attached to the INDEX file and that Unit 4 is attached to file CROPDATA.

#### 3.2.2.4 Outputs

After ejecting one page on the printer, INIT writes a heading for all output produced by subsequent subroutines and main program statements.

#### 3.2.2.5 Description

INIT opens file CROPDATA for direct access, reads part of the INDEX file into the COMMON block INDEX and part into COMMON block CLOSE, prints a heading, creates COMMON block DATA, and initializes variables in COMMON block TALK, which is used in SWAP.

#### 3.2.2.6 Flowchart

Next page.

#### 3.2.2.7 Listing

Follows flowchart.

~~3-12~~



~~SECRET~~

RUN NO. 73

DATE 05/04/77

TIME 1820

LISTING OF MODULE INIT

DESCRIPTION

INITS AUTOMATION FILE ACCESS

MASTER FILE W.ECC.CCEA.GVM.MASTER

ADDED TO MASTER 07/25/76

LAST DATE COPIED NONE

LAST UPDATE 07/25/76 1809

PASSWORD DGLJ

PROGRAMMER VONHOLT

LANGUAGE FOR

PROC PARAMETER SNOJCL

SUBROUTINE INIT

REAL\*8 BUFFER(128),BUF2(128)

INTEGER WMO(511),STATN(511,3)

INTEGER ACCESS,CHANGE,PAGE1,PAGE0

INTEGER\*2 NUMSTA,FILLER,CCDATA(2,16,16)

COMMON /TALK/INCORE,ACCESS,CHANGE,PAGE1,PAGE0

COMMON /INDEX/NUMSTA,FILLER,WMO

COMMON /DATA/CCDATA

COMMON /CLOSE/STATN

EQUIVALENCE (CCDATA(1,1,1),BUFFER(1))

EQUIVALENCE (NUMSTA,BUF2(1))

C  
H  
1001  
REWIND 2  
WRITE(6,15)  
15  
S  
FORMAT(1,1) CCEA PGM CC76-COU-02-02 CROP CALENDAR DATA CAPTURE  
S  
VERSION OF JULY, 1976  
INCORE = -1  
ACCESS=0  
CHANGE=0  
PAGE1 = 0  
PAGE0=0  
C  
RETURN  
END00000010  
00000020  
00000030  
00000040  
00000050  
00000060  
00000070  
00000080  
00000090  
00000100  
00000110  
00000120  
00000130  
00000140  
00000150  
00000160  
00000170  
00000180  
00000190  
00000200  
00000210  
00000220  
00000230  
00000240  
00000250  
00000260  
00000270  
00000280

07/25/76

07/25/76  
07/25/76ORIGINAL PAGE IS  
OF POOR QUALITY

### 3.2.3 SUBROUTINE SWAP

SWAP is the entry point used by all crop calendar preparation routines to read from or write to file CROPDATA. The 'in' version of SWAP reads from CROPDATA, and the 'out' version writes to CROPDATA.

#### 3.2.3.1 Linkages

SWAP uses the COMMON blocks TALK, INDEX, and DATA, which were declared by subroutine INIT.

#### 3.2.3.2 Interfaces

INIT must be called before SWAP.

#### 3.2.3.3 Inputs and Outputs

INIT assumes Unit 4 is attached to file CROPDATA.

ACTION	if 1, then read from CROPDATA, with TEMPS and JULIAN as outputs to the main program; if 2, then write to CROPDATA, with TEMPS and JULIAN as inputs from the main program.
WMONUM	station number of requested station.
TEMPS	array of 14 days' maximum and minimum temperatures associated with WMONUM.
JULIAN	first date of the 14-day period.
RCODE	if -1, then ACTION was other than 1 or 2; if 0, then transaction was successful; if 10, then WMONUM was not in the INDEX file.

#### 3.2.3.4 Description

SWAP maintains a BUFFER in COMMON block DATA to contain a 16-station block of file CROPDATA; it also maintains a list of crop calendar stations in COMMON INDEX. The list of WMO numbers is in the same order as the data on CROPDATA. To access a WMO station, SWAP sequentially searches INDEX for the requested number. If the number is found, its position in INDEX is used to determine the block address (i.e., in which 16-station block of CROPDATA the station is located). If the station number is not found, RCODE is set to 10. At this point it is determined whether the requested block is in core. If the block is in core, it is read from or written

to; if the necessary block is not in core, it is swapped with the block currently in core.

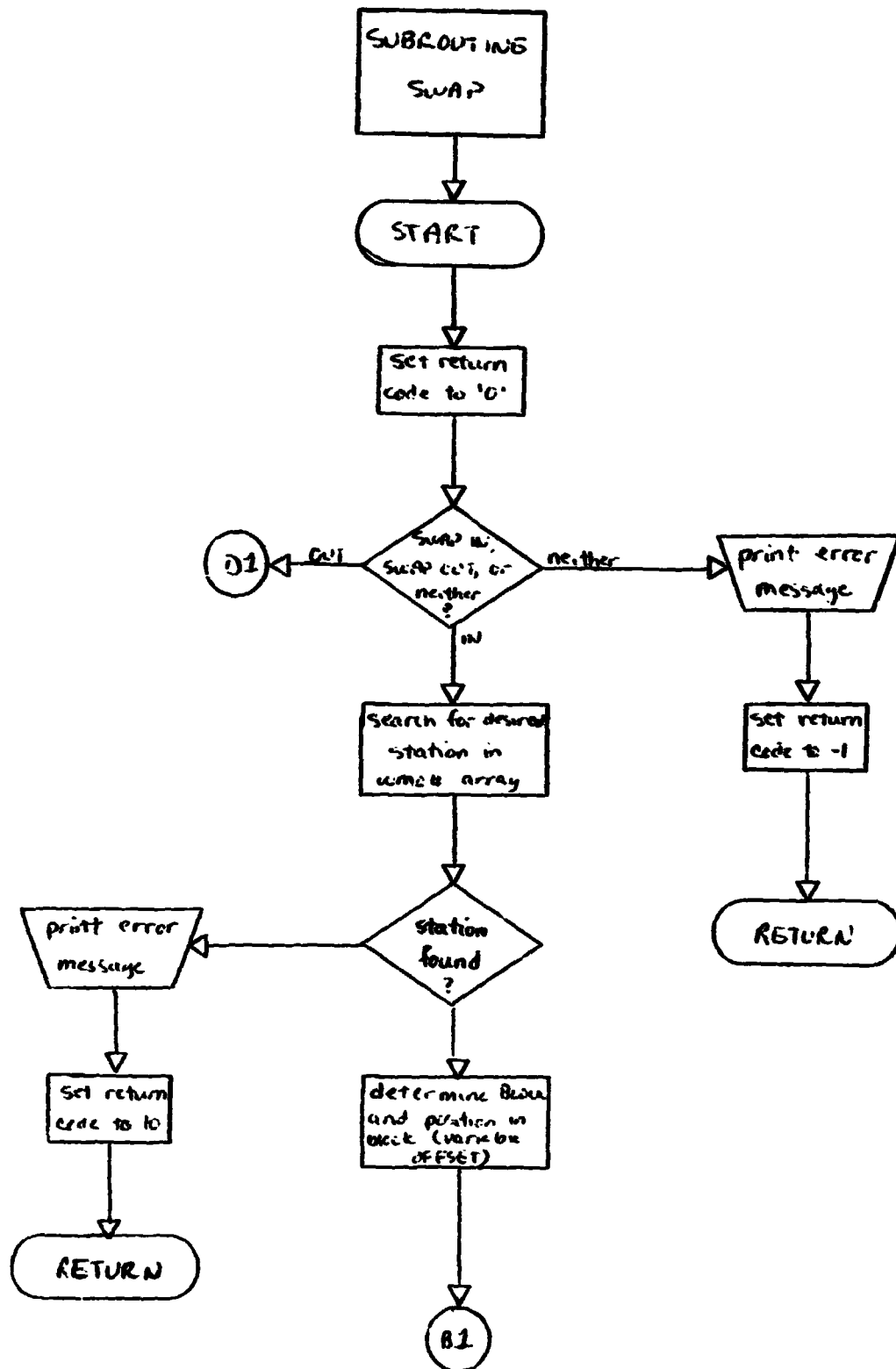
#### 3.2.3.5 Flowchart

Next page.

#### 3.2.3.6 Listing

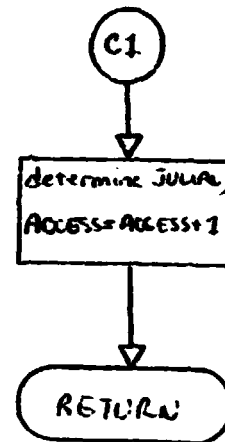
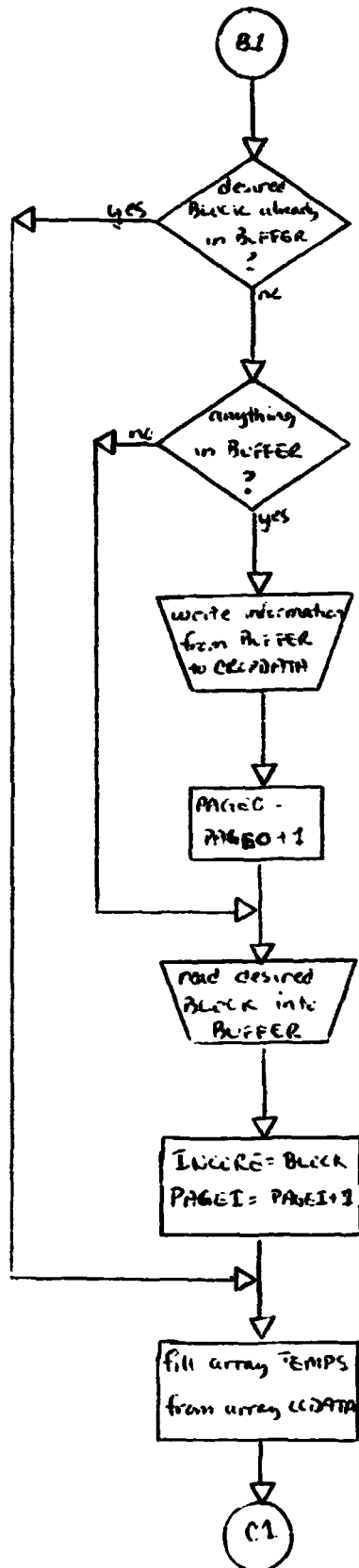
Follows flowchart.



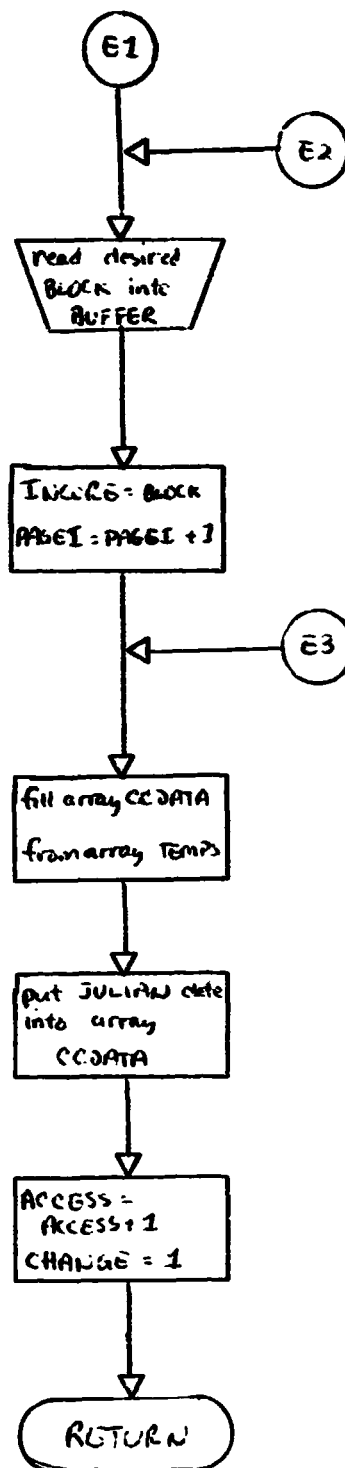
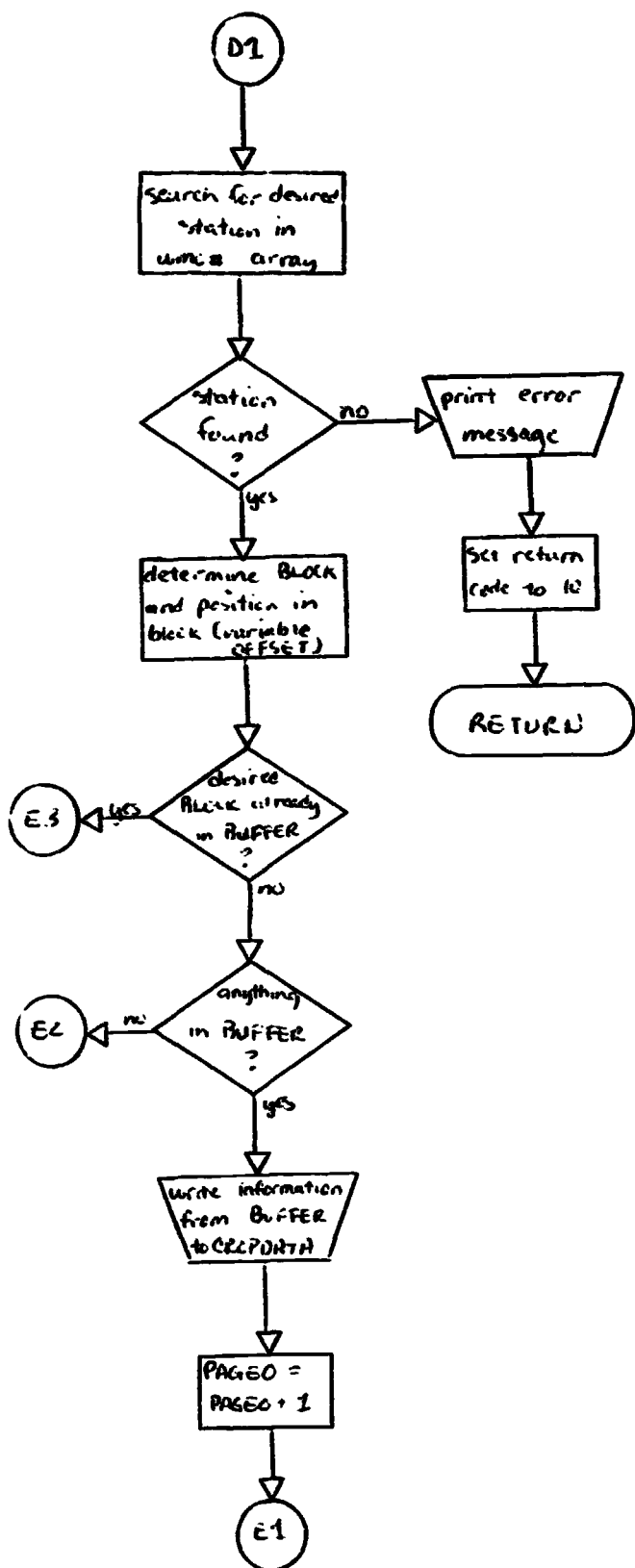


~~3-15~~

# SUBROUTINE SWAP, CONT



# SUBROUTINE SWAP, CONT



RUN NO. 73

DATE 05/04/77

TIME 1820

## LISTING OF MODULE SWAP

DESCRIPTION SWAPS MET DATA FOR AUTOMATION

MASTER FILE W.EDS.CCEA.GVM.MASTER

ADDED TO MASTER 07/25/76

LAST DATE COPIED NONE

LAST UPDATE 07/25/76 1719

PASSWORD PDXS

PROGRAMMER VONHOLT

LANGUAGE FOR

PROC PARAMETER \$NOJCL

SUBROUTINE SWAP(ACTION,WMONUM,TFMPS,JULIAN,IER)

REAL\*8 BUFFER(128)

INTEGER ACTION,WMONUM,TFMPS(14,2),WMO(511),POSIT

INTEGER ACCESS,CHANGE,PAGEI,PAGEO

INTEGER BLOCK,OFFSET

INTEGER\*2 NUMSTA,FILLER,CCDATA(2,16,16)

COMMON /TALK/INCORE,ACCESS,CHANGE,PAGEI,PAGEO

COMMON /INDEX/NUMSTA,FILLER,WMO

COMMON /DATA/CCDATA

EQUIVALENCE(CCDATA(1,1,1),BUFFER(1))

C

IER = 0

IF ((ACTION.LT.1).OR.(ACTION.GT.2)) GO TO 3000

GO TO (1000,2000), ACTION

C\*\*\* SWAP IN

DO 800 J = 1, NUMSTA

IF (WMO(J).EQ.WMONUM) GO TO 900

800 CONTINUE

GO TO 4000

900 POSIT = J

BLOCK = (POSIT-1) / 16 + 1

OFFSET=POSIT-(BLOCK-1)\*16

IF (INCORE.EQ.BLOCK) GO TO 1050

IF (CHANGE) 1040, 1040, 1030

1030 WRITE(4,INCORE,1001) BUFFER

1001 FORMAT(12HAR)

PAGEO=PAGEO+1

1040 HEAD(4,BLOCK,1001) BUFFER

INCORE=BLOCK

PAGEI=PAGEI+1

C

1050 DO 1060 J = 1, 14

TFMPS(J,1)=CCDATA(1,J+2,OFFSET)

1060 TFMPS(J,2)=CCDATA(2,J+2,OFFSET)

JULIAN=CCDATA(1,1,OFFSET)

ACCESS=ACCESS+1

RETURN

C\*\*\* SWAP OUT

2000 DO 1800 J = 1, NUMSTA

IF (WMO(J).EQ.WMONUM) GO TO 1900

1800 CONTINUE

GO TO 4000

1900 POSIT=J

00000010

00000020

00000030

00000040

00000050

00000060

00000070

00000080

00000090

00000100

00000110

00000120

00000130

00000140

00000150

00000160

00000170

00000180

00000190

00000200

00000210

00000220

00000230

00000240

00000250

00000260

00000270

00000280

00000290

00000300

00000310

00000320

00000330

00000340

00000350

00000360

00000370

00000380

00000390

00000400

00000410

00000420

00000430

07/25/76

07/25/76

ORIGINAL PAGE IS  
OF POOR QUALITY

RUN NO. 73      DATE 05/04/77      TIME 1820

LISTING OF MODULE SWAP

	RLOCK = (POSIT-1) / 16 + 1	00000440	
	OFFSET=POSIT-(RLOCK-1)*16	00000450	
	IF (INCORE.EQ.HLOCK) GO TO 2050	00000460	
	IF (CHANGE) 2040, 2040, 2030	00000470	
2030	WRITE(4,INCORE,1001)BUFFER	00000480	
	PAGE0=PAGE0+1	00000490	
2040	READ(4,RLOCK,1001)BUFFER	00000500	
	INCORE=RLOCK	00000510	
	PAGE1=PAGE1+1	00000520	
C		00000530	
2050	DO 2060 J = 1, 14	00000540	
	CCDATA(1,J+2,OFFSET)=TEMPS(J,1)	00000550	
2060	CCDATA(2,J+2,OFFSET)=TEMPS(J,2)	00000560	
	CCDATA(1,1,OFFSET)=JULIAN	00000570	
	ACCESS=ACCESS+1	00000580	
	CHANGE=1	00000590	
	RETURN	00000600	
C		00000610	
3000	WRITE(6,3010)ACTION	00000620	
3010	FORMAT('0INVALID ACTION CODE:',I7)	00000630	
	CALL ERNTRA	00000640	
	STOP	00000650	
C		00000660	
4000	IER = 1	00000670	
	RETURN	00000680	
C		00000690	
	END	00000700	

07/25/76  
07/25/76

### 3.2.4 SUBROUTINE TERM

TERM is used by all crop calendar preparation routines to close the weather file and print statistics accumulated during the running of SWAP.

#### 3.2.4.1 Linkages

TERM utilizes COMMON blocks TALK and DATA.

#### 3.2.4.2 Interfaces

SWAP must be called before TERM. TERM is the last subroutine called before the end of any crop calendar preparation program.

#### 3.2.4.3 Inputs

TERM uses COMMON blocks TALK and DATA.

#### 3.2.4.4 Outputs

TERM may write a block to CROPDATA, assumed on Unit 4. The subroutine also prints out SWAP 'IN'/SWAP 'OUT' statistics and a closing message.

#### 3.2.4.5 Description

If a 16-station block remains in BUFFER, TERM writes the block to CROPDATA. Statistics accumulated during the execution of SWAP and a closing message are written to the printer.

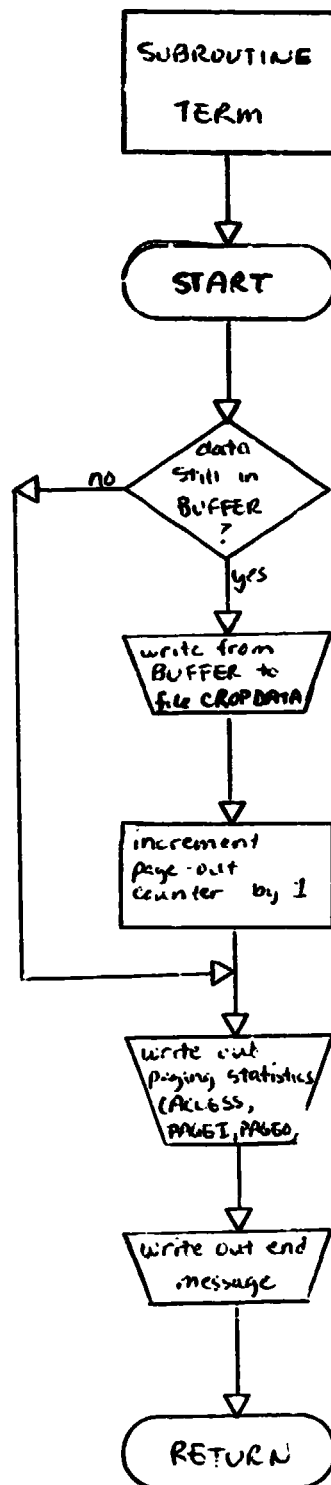
#### 3.2.4.6 Flowchart

Next page.

#### 3.2.4.7 Listing

Follows flowchart.

~~3-20~~



~~3.23~~

RUN NO. 73      DATE 05/04/77      TIME 1820

LISTING OF MODULE TERM

DESCRIPTION      TERM AUTOMATION FILE

MASTER FILE      W.EDS.CCEA.GVM.MASTER  
ADDED TO MASTER      07/25/76  
LAST DATE COPIED      NONE  
LAST UPDATE      NONE

PASSWORD      HLSK  
PROGRAMMER      VONHOLT  
LANGUAGE      FOR  
PROC PARAMETER      \$NOJCL

SUBROUTINE TERM  
REAL\*8 BUFFER(128)  
INTEGER ACCESS.CHANGE,PAGE1,PAGE0  
COMMON /TALK/INCORE,ACCESS.CHANGE,PAGE1,PAGE0  
COMMON /DATA/BUFFER

00000010  
00000020  
00000030  
00000040  
00000050  
00000060  
00000070  
00000080  
00000090  
00000100  
00000110  
00000120  
00000130  
00000140  
00000150  
00000160  
00000170

C

IF (CHANGE) 200 , 200 , 100  
100 WRITE(4,INCORE,110)BUFFER  
110 FORMAT(128A8)  
PAGE0=PAGE0+1  
200 WRITE(6,210)ACCESS,PAGE1,PAGE0  
210 FORMAT(1-\*,PAGING STATISTICS \*\*/,,'-ACCESSES',115,/,  
\* '-PAGE INS',115,/,,'-PAGE OUTS',114)  
1 WRITE(6,1)  
1 FORMAT(1-/,,'-\*,/,,'-\*\* END OF CAPTURE ROUTINE EXECUTION \*\*)  
RETURN  
END

ORIGINAL PAGE IS  
OF POOR QUALITY



### 3.2.5 PROGRAM CLEAR

CLEAR resets all entries in CROPDATA to missing values (9999) and calculates the starting Julian date for the 14-day period about to be processed.

#### 3.2.5.1 Linkages

CLEAR calls subroutines INIT, SWAP, AND TERM and uses COMMON block INDEX.

#### 3.2.5.2 Interfaces

CLEAR is the first crop calendar preparation routine to be run for any given 14-day period.

#### 3.2.5.3 Inputs

The INDEX file is on Unit 2, CROPDATA is on Unit 4, and optional card input (date - reset value) on Unit 5.

#### 3.2.5.4 Outputs

The CROPDATA file is initialized.

#### 3.2.5.5 Description

CLEAR calls INIT to load COMMON block INDEX and open CROPDATA to direct access. An attempt is made to read a date-reset value from Unit 5; if no value is present, the reset value defaults to 14 days. The starting date of the 14-day period to be processed is reset and all temperatures in CROPDATA are set to '9999', using SWAP. The file is then closed by subroutine TERM.

#### 3.2.5.6 Flowchart

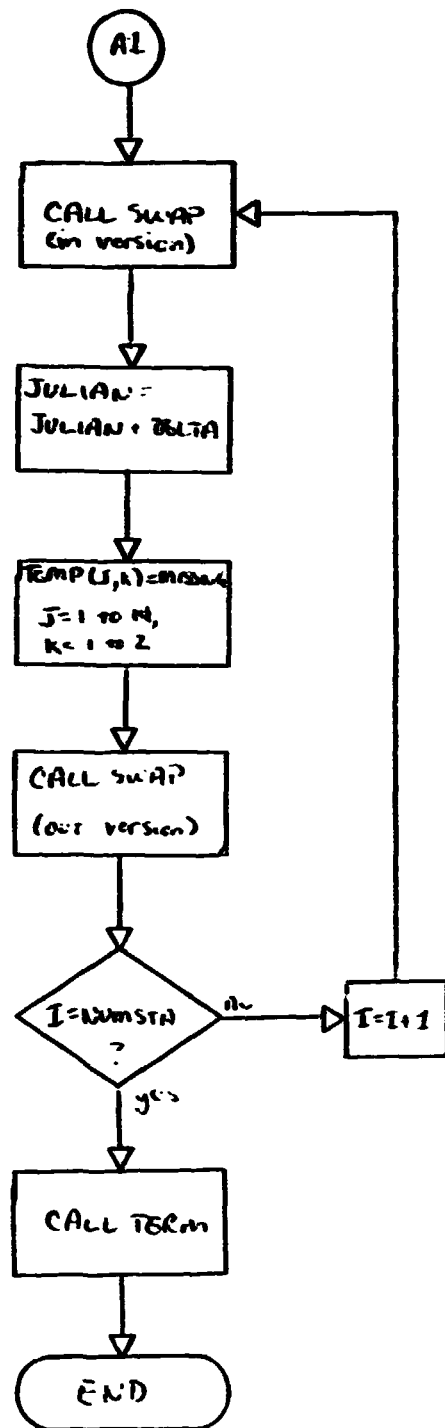
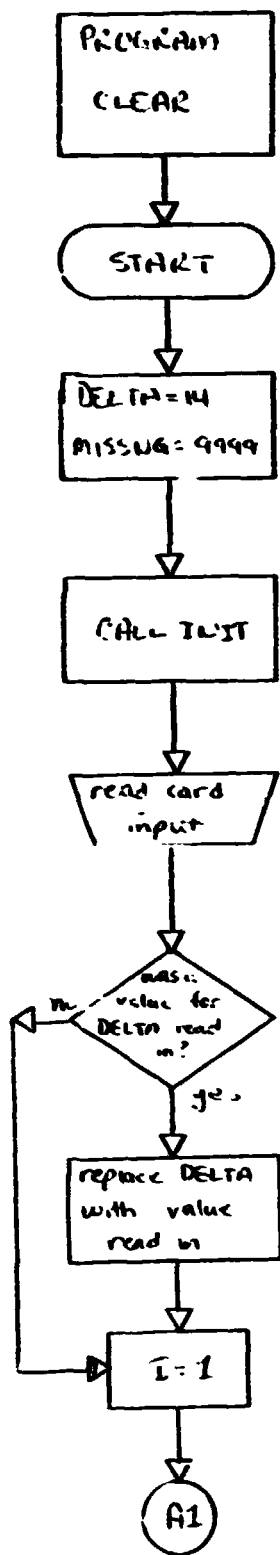
Next page.

#### 3.2.5.7 Listing

Follows flowchart.

~~9999~~

35



## LISTING OF MODULE CLEAR

MASTER FILE	W. ENS. CCEA. GVM. MASTER
ADDED TO MASTER	07/25/76
LAST DATE COPIED	NONE
LAST UPDATE	07/25/76 2129

PASSWORD	SLFH
PROGRAMMER	VONHOLT
LANGUAGE	FOR
PROC PARAMETER	SNOJCL

```

INTEGER WMO(511),TEMPS(14,2);JUCIAN,IER,IN/T/,OUT/2/,MISSNG/9999/
INTEGER DELTA/14/
INTEGER*2 NUMSTA,FILLER
COMMON /INDEX/NUMSTA,FILLER,WMO

```

[illegible]

**07/25/76**

```
CALL INIT
READ(5,*,END=2) DELTA
```

2 CONTINUE

DO 10 I = 1, NUMSTA

CALL SWAP (IN, WMC (I),

JULIAN = JULIAN + DELTA

[illegible]

TEMPS(J,K) = MISSING

CALL SWAP(OUT,WMO(I),TEMPS,JULIAN,IER)

10 CONTINUE  
CALL TER

**CALL TERM**

STOP  
END

ORIGINAL PAGE IS  
DE POOR QUALITY

### 3.2.6 JCL FILE CLEAR

Executing JCL file CLEAR invokes the CLEAR program.

#### 3.2.6.1 Linkages

None.

#### 3.2.6.2 Interfaces

None.

#### 3.2.6.3 Inputs

See Sections 3.2.5.3 and 4.2.2.1.

#### 3.2.6.4 Outputs

None.

#### 3.2.6.5 Description

JCL file CLEAR binds the files referenced by program CLEAR to that program and requests that the program be executed.

#### 3.2.6.6 Listing

```
//CLEAR//PROC
//CLEAR//EXEC//PGM= CLEAR
//STEPLIB//DD//DSN=W.EDS.CCEA.PHASE3.LOAD.DISP=SHR
//FT02F001//DD//DSN=W.EDS.CCEA. { GROUP I } .INDEX,DISP=SHR
//FT04F001//DD//DSN=W.EDS.CCEA. { GROUP I } .CROPDATA,DISP=SHR
//FT06F001//DD//SYSOUT=A
//FT05F001//DD//DDNAME=SYSIN
//P//PEND
```

~~3-28~~

### 3.2.7 PROGRAM CAPTURE

CAPTURE reads the tape and disk temperature data files built at CCEA/Washington and loads them into CROPDATA.

#### 3.2.7.1 Linkages

CAPTURE calls subroutines INIT, CVB, SWAP, and TERM.

#### 3.2.7.2 Interfaces

CLEAR must be run before CAPTURE.

#### 3.2.7.3 Inputs

CAPTURE assumes the INDEX file is on Unit 2, the tape or disk data file on Unit 3, CROPDATA on Unit 4, and an optional card input file on Unit 5. This latter file is used with North American data only; it contains the Julian date of the first day of the month being processed, and the new year if the 2-week period spans two years.

#### 3.2.7.4 Outputs

CAPTURE loads CROPDATA with temperature data for the 14-day period being processed.

#### 3.2.7.5 Description

CAPTURE reads the eleventh record from Unit 3 to determine if the data is "North American" or "non-North American". Unit 3 is rewound and INIT is called. For North American data, the Julian date of the first day of the month is read from Unit 5 and the array JULIAN is filled. North American data also requires that CAPTURE call subroutine CVB. For each new station SWAP(in) is called; CAPTURE determines what portion of the 14-day crop calendar is being accessed by this run of the program. CAPTURE makes the necessary adjustments if the 2-week period spans two years, including reading another card from Unit 5 for North American data to obtain the new year. Using SWAP(out), CAPTURE updates the CROPDATA file with the new weather data. After all stations have been processed, TERM is called and processing ceases.

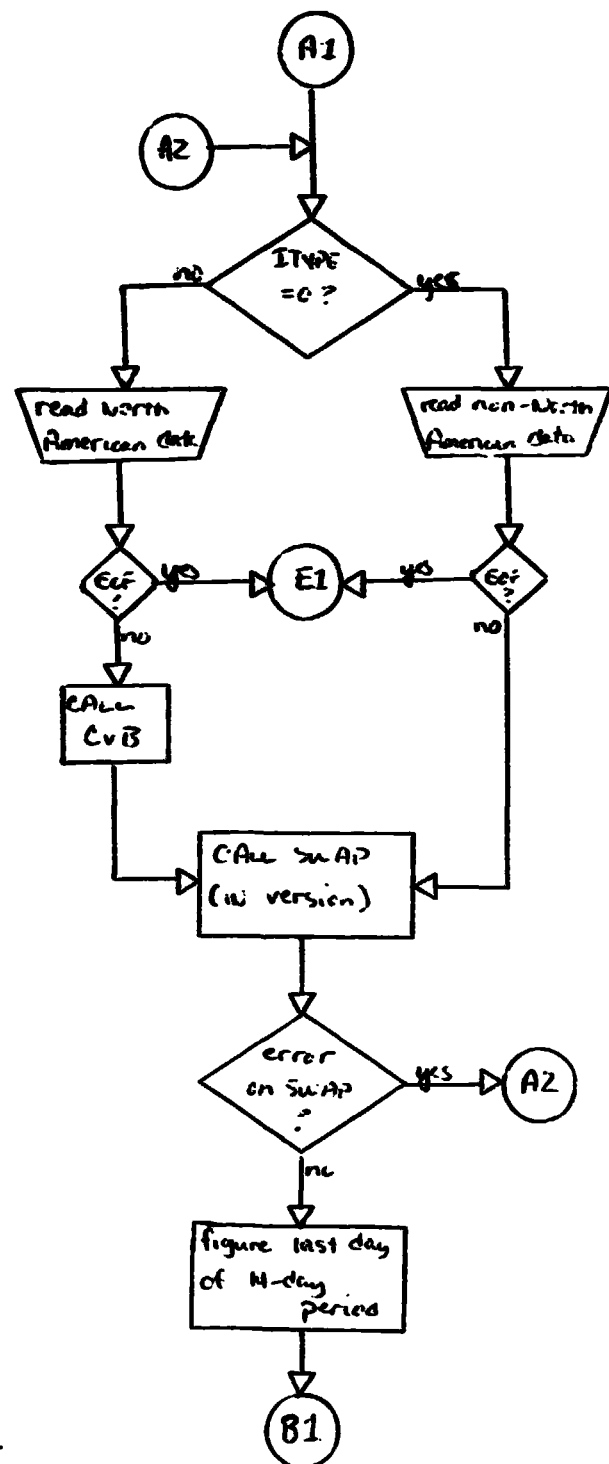
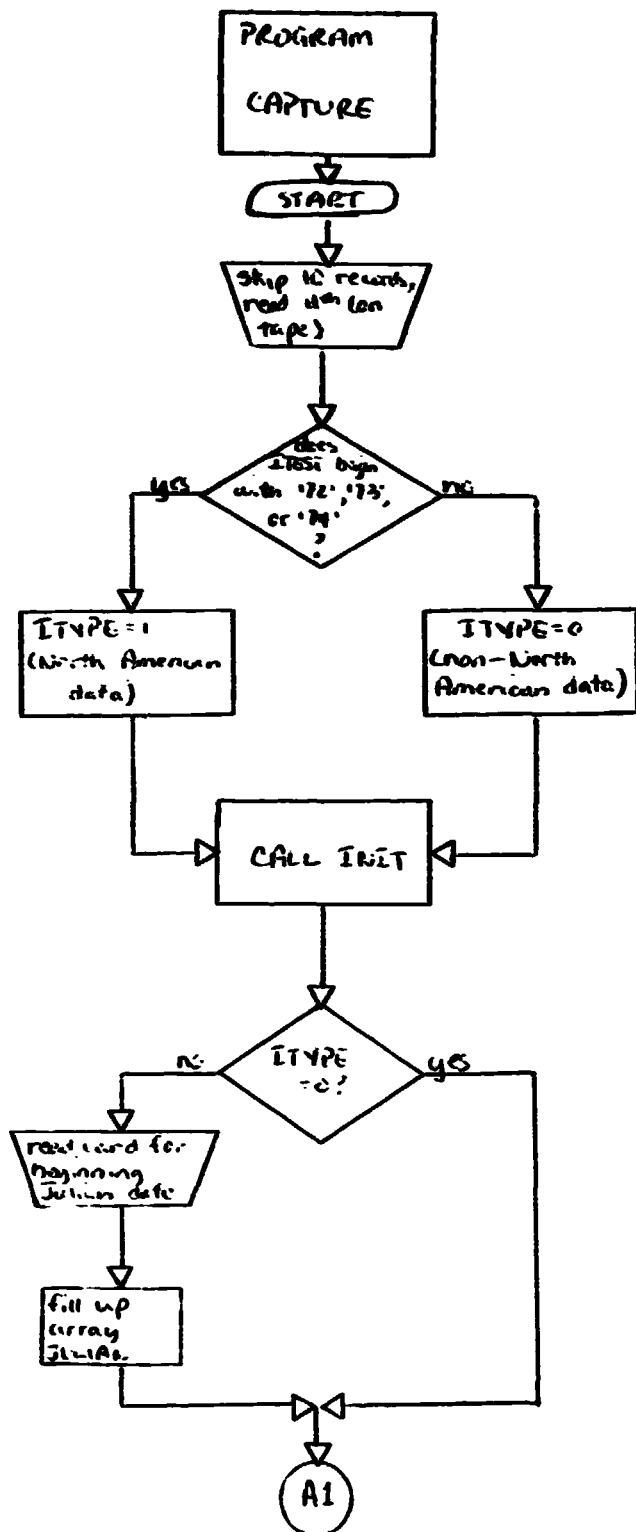
~~300~~

**3.2.7.6 Flowchart**

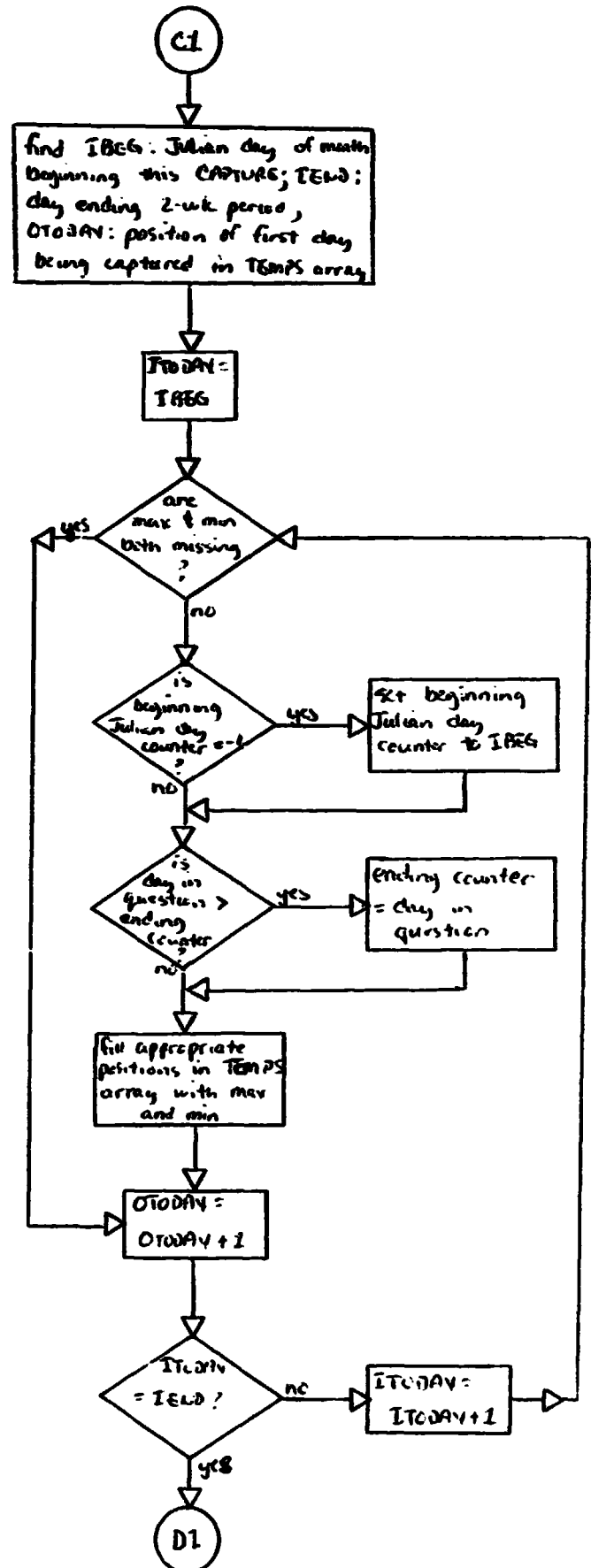
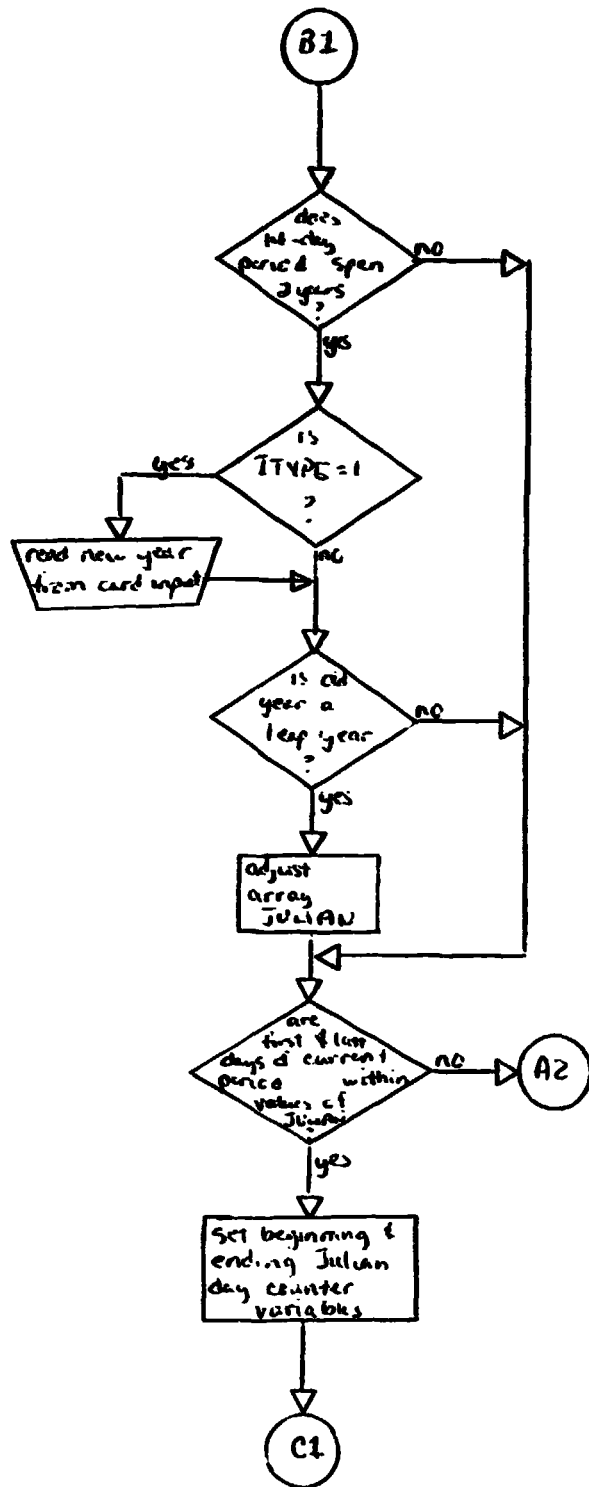
Next page.

**3.2.7.7 Listing**

Follows flowchart.

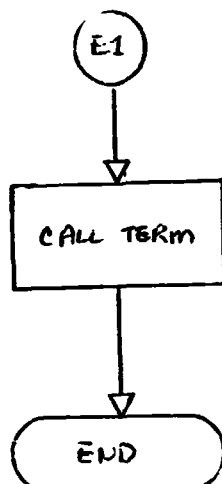
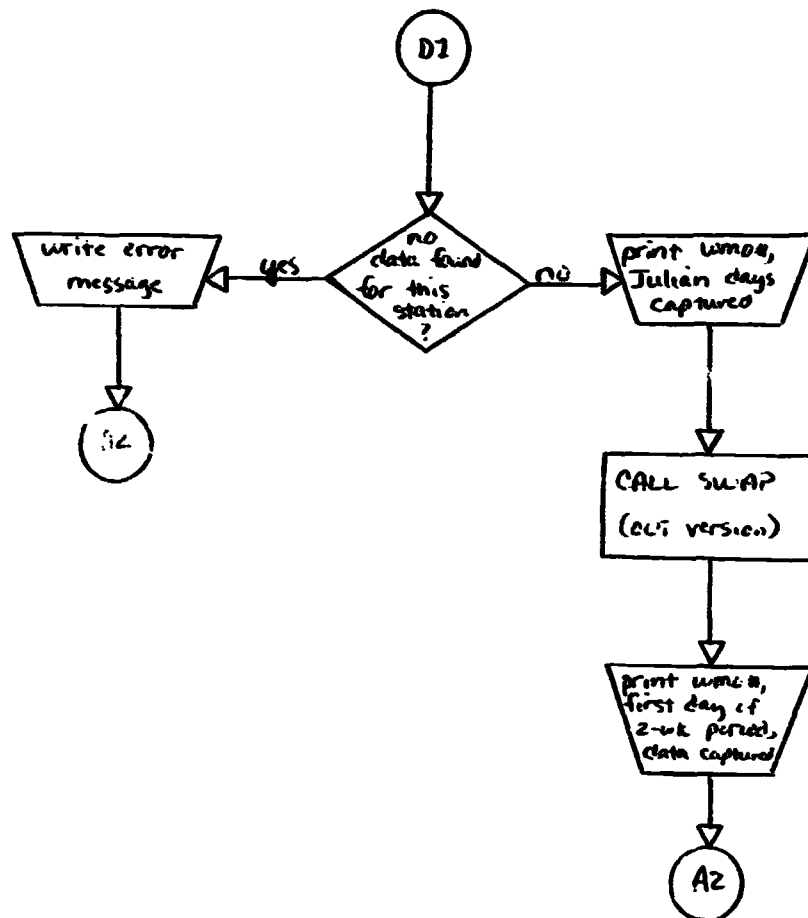


# PROGRAM CAPTURE, CONT





# PROGRAM CAPTURE, CONT





RUN NO. 70

DATE 04/28/77

TIME 1633

### LISTING OF MODULE CAPTURE

Line	Code	Statement	Address	Date
1	C	CHECK FOR LEAP YEAR	00000440	01/10/77
		CHECK=(YR-1.0)/4	00000450	01/10/77
		ICHECK=CHECK	00000460	01/10/77
		IF (CHECK.EQ.ICHECK) J=1	00000470	01/10/77
		IF (J.EQ.1 .AND. LDAY.LE.366) GO TO 35	00000480	01/10/77
3	C	ADJUST JULIAN DATE	00000490	01/10/77
		DO 36 I=1,31	00000500	01/10/77
		JULIAN(I)=JULIAN(I)+365+J	00000510	01/10/77
36		CONTINUE	00000520	01/10/77
35		CONTINUE	00000530	01/10/77
		IF ((LDAY.LT.JULIAN(I)).OR.(JDAY.GT.JULIAN(I)+30)) GO TO 20	00000540	01/10/77
		VREG = -1	00000550	01/10/77
		VEND = -1	00000560	01/10/77
		IF (JDAY.LE.JULIAN(I)) GO TO 40	00000570	01/10/77
		IHEG = JDAY + 1 - JULIAN(I)	00000580	07/25/76
		OTODAY = 1	00000590	07/25/76
		IEND = MIN(31,IHEG+13)	00000600	07/25/76
		GO TO 50	00000610	
40		IHEG = 1	00000620	
		OTODAY = JULIAN(I) + 1 - JDAY	00000630	
		IEND = 15 - OTODAY	00000640	07/25/76
50		DO 60 ITODAY = IHEG, IEND	00000650	
		IF ((LOW(ITODAY).EQ.9999).AND.(HIGH(ITODAY).EQ.9999)) GO TO 60	00000660	
		IF (VREG.EQ.-1) VREG = JULIAN(ITODAY)	00000670	
		IF (JULIAN(ITODAY).GT.VEND) VEND=JULIAN(ITODAY)	00000680	
		TEMPS(OTODAY,1) = LOW(ITODAY)	00000690	
		TEMPS(OTODAY,2) = HIGH(ITODAY)	00000700	
60		OTODAY = OTODAY + 1	00000710	
		IF (VREG.EQ.-1) GO TO 65	00000720	07/25/76
		WRITE(5,53) WMONNM,VREG,VEND	00000730	
53		FORMAT(1X,15,'(,13,'-',13,')')	00000740	07/25/76
		CALL SWAP(OUT.WMONNM,TEMPS,JDAY,IFW)	00000750	
		WRITE(1,54) WMONNM,JDAY,(TEMPS(J,2),J=1,14),WMONNM,JDAY,	00000760	
		(TEMPS(J,1),J=1,14)	00000770	07/25/76
54		FORMAT(15,' MX',14,' 14',1416,'/,15,' MN',14,' 14',1416)	00000780	07/25/76
		GO TO 20	00000790	
65		WRITE(5,67) WMONNM	00000800	07/25/76
67		FORMAT(16,' ( NONE )')	00000810	07/25/76
		GO TO 20	00000820	07/25/76
100		CALL TERM	00000830	
		STOP	00000840	07/25/76
		END	00000850	07/25/76

### 3.2.8 JCL FILE CAPTURE

Executing JCL file CAPTURE involves the CAPTURE program.

#### 3.2.8.1 Linkages

None.

#### 3.2.8.2 Interfaces

None.

#### 3.2.8.3 Inputs

Direct card inputs to this file are necessary only with North American data. See Sections 3.2.7.3 and 4.2.2.2.

#### 3.2.8.4 Outputs

None.

#### 3.2.8.5 Description

JCL file CAPTURE binds the files referenced by program CAPTURE to that program and requests invocation of program CAPTURE.

#### 3.2.8.6 Listing

```
//CAPTUREØPROCØUNIT=,VOL=,DSN='W.EDS.CCEA.DAYDATA'  
//CAPTUREØEXECØPGM=CAPTURE  
//STEPLIBØDDØDSN=W.EDS.CCEA.PHASE3.LOAD,DISP=SHR  
//FT01F001ØDDØSYSOUT=A,DCB=(LRECL=100,BLKSIZE=1500,RECFM=FB)  
//FT02F001ØDDØDSN=W.EDS.CCEA.{GROUPI}.INDEX,DISP=SHR  
//FT03F001ØDDØDSN=&DSN.,DISP=SHR,UNIT=&UNIT.,VOL=SER=&VOL,  
//ØØØØLABEL=(,,IN)  
//FT04F001ØDDØDSN=W.EDS.CCEA.{GROUPI}.CROPDATA,DISP=SHR  
//FT06F001ØDDØSYSOUT=A  
//ØØØØØPEND
```

~~36~~

### 3.2.9 SUBROUTINE CVB

CVB is called by CAPTURE to convert WMO numbers from zoned decimal to binary (North American data only).

#### 3.2.9.1 Linkages

None.

#### 3.2.9.2 Interfaces

None.

#### 3.2.9.3 Inputs and Outputs

FROM	the input WMO number (zoned decimal)
NCHAR	number of bytes in FROM (in this case, NCHAR=5)
TO	the output WMO number (binary)

#### 3.2.9.4 Description

North American WMO numbers exist as zoned decimal and need to be converted to binary before processing can take place. CVB accomplishes this conversion.

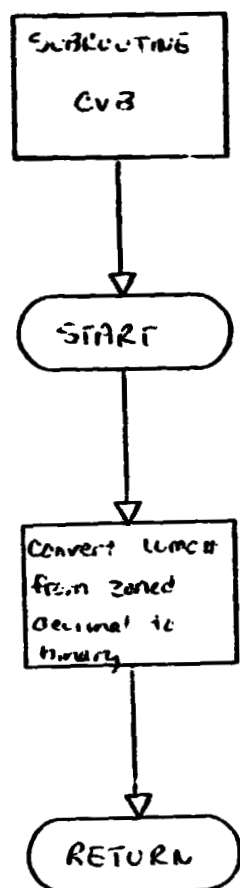
#### 3.2.9.5 Flowchart

Next page.

#### 3.2.9.6 Listing

Follows flowchart.

~~3-25~~



~~3-06~~

38

```

//STEP1 EXEC NASMHC
//ASM.SYSIN DD *
CVR TITLE '- CONVERTS FROM ZONED DECIMAL TO BINARY'
CVR CSECT
CVR SPACE
*****
SUBROUTINE CVR(FROM,NCHAR,TO)

PURPOSE  CONVERT NCHAR CHARACTERS(BYTES) FROM ZONED DECIMAL
          (EBCDIC) TO BINARY, STORING FULLWORD RESULT AT
          LOCATION TO.  NCHAR MUST BE BETWEEN 1 AND 16,
          INCLUSIVE.

LANGUAGE ASSEMBLER

WRITTEN BY G VON HOLT, JULY 1976
          FORTRAN X + H COMPATIBLE
*****
SPACE 2
USING 9,15
H BEGIN
DC AL1(3)
DC CL3,CVH
BEGIN STM 14,12,12(13)
LM 3,5,0(1)
L 4,0(4)
HCTR 4,0
XC DBL,DBL
EX 4,PACK
CVR 6,DBL
ST 6,0(5)
LM 14,12,12(13)
HR 14
SPACE 2

ADDRESSABILITY
BRANCH AROUND NAME
LENGTH OF NAME
NAME OF SUBROUTINE
SAVE CALLER'S REGISTERS
GET ADDRESS OF PARAMS.
GET NUMBER OF CHARACTERS
SUBTRACT 1 FOR EX INST.
CLEAR RESULT FIELD
CONVERT TO PACKED DECIMAL
CONVERT PACKED DEC TO BINARY
AND PLACE IN RESULT AREA
RESTORE CALLER'S REGISTERS
RETURN

OBJECT OF EX INST.

PACK DBL+4(4),0(0,3)
SPACE 2
CONVERTS TO PACKED DECIMAL

SPACE FOR PACKED DECIMAL NUMBER

DBL DS 0
SPACE 2
END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

### 3.2.10 PROGRAM PRINT

PRINT lists the contents of the CROPDATA file. It may be run at any time.

#### 3.2.10.1 Linkages

PRINT calls INIT, SWAP, and TERM and uses COMMON block INDEX.

#### 3.2.10.2 Interfaces

None.

#### 3.2.10.3 Inputs

PRINT assumes the INDEX file is on Unit 2 and the CROPDATA file is on Unit 4.

#### 3.2.10.4 Outputs

PRINT lists each of the stations on file INDEX and the corresponding temperature data from file CROPDATA.

#### 3.2.10.5 Description

PRINT calls INIT and proceeds to call SWAP for each station on the INDEX file, printing the station number, starting Julian date of the current 2-week period, and the maximum and minimum temperatures in the 2-week period. After information for all stations on INDEX has been listed, TERM is called.

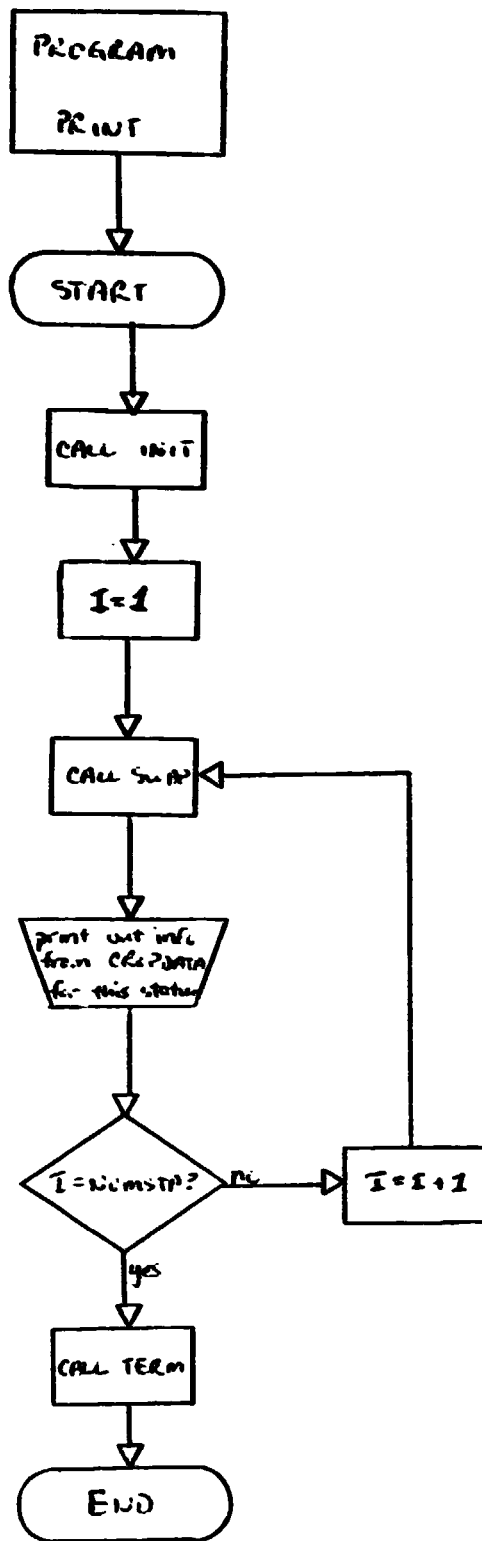
#### 3.2.1.6 Flowchart

Next page.

#### 3.2.10.7 Listing

Follows flowchart.





~~2009~~  
H/

RUN NO. 73

DATE 05/04/77

TIME 1820

LISTING OF MODULE PRINT

DESCRIPTION

PRINTS OUT AUTOMATION FILE

MASTER FILE

W.EDS.CCEA.GVH.MASTER

ADDED TO MASTER

07/25/76

LAST DATE COPIED

NONE

LAST UPDATE

07/25/76 2014

PASSWORD

HXFH

PROGRAMMER

VONHOLT

LANGUAGE

FOR

PROC PARAMETER

\$NOJCL

INTEGER WMO(511),IN/1,TEMPS(14,2),JULIAN,IER

00000010

INTEGER\*2 NUMSTA,FILLER

00000020

COMMON /INDEX/NUMSTA,FILLER,WMO

00000030

CALL INIT

00000040

WRITE(6,5)

00000050

5 FORMAT('1')

00000060

DO 10 I = 1, NUMSTA

00000070

CALL SWAP(IN,WMO(I),TEMPS,JULIAN,IER)

00000080

10 WRITE(6,20)WMO(I),JULIAN,(TEMPS(J,2),J=1,14),

00000090

WMO(I),JULIAN,(TEMPS(J,1),J=1,14)

00000100

20 FORMAT('6.1 MAX:15.2X.14 6.1,

00000110

6.1 MIN:15.2X.14 6.1)

00000120

WRITE(6,5)

00000130

CALL TERM

00000140

STOP

00000150

END

00000160

07/25/76

ORIGINAL PAGE IS  
OF POOR QUALITY

### 3.2.11 JCL FILE LIST

**Executing JCL file LIST invokes the PRINT program.**

### 3.2.11.1 Linkages

**See Section 3.2.10.1.**

### 3.2.11.2 Interfaces

**None.**

### 3.2.11.3 Inputs

**None.**

#### 3.2.11.4 Outputs

**None.**

#### 3.2.11.5 Description

**JCL file LIST binds the files referenced by program PRINT to that program and requests execution of program PRINT.**

### 3.2.11.6 Listing

```
//LISTXXXXPROC
//LISTXXXXEXECXXXXPGM=PRINT
//STEPLIBXXXXDDXXXXDSN=W.EDS.CCEA.PHASE3.LOAD,DISP=SHR
//FT02F001XXXXDDXXXXDSN=W.EDS.CCEA.{GROUP I}.INDEX,DISP=SHR
//FT04F001XXXXDDXXXXDSN=W.EDS.CCEA.{GROUP I}.CROPDATA,DISP=SHR
//FT06F001XXXXDDXXXXSYSOBT=A
//XXXXXXPEND
```

### 3.2.12 PROGRAM EDITOR

EDITOR checks for inverted temperatures of missing data and attempts to replace bad values by substituting data from a given station's three closest neighbor stations.

#### 3.2.12.1 Linkages

EDITOR calls, INIT, SWAP, CHECK, REPLAC, and TERM and uses COMMON blocks INDEX, VAL1, and CLOSE.

#### 3.2.12.2 Interfaces

EDITOR is not executed until all runs of CAPTURE have been completed and is run before the CROPCALN program is executed.

#### 3.2.12.3 Inputs

EDITOR assumes that the INDEX file is on Unit 2, CROPDATA on Unit 4, and card input on Unit 5, containing the variables IP, IC, IE, and IS.

These four variables attain the following values:

- IP = 0 no listing
- = 1 list the file CROPDATA
- IC = 0 do not call CHECK or REPLAC
- = 1 call CHECK but do not print results
- = 2 call CHECK and print results
- IE = 0 do not call REPLAC
- = 1 call REPLAC
- IS = 0 do not rewrite CROPDATA
- = 1 rewrite CROPDATA

#### 3.2.12.4 Outputs

EDITOR produces corrections for CROPDATA and writes messages to the printer.

#### 3.2.12.5 Description

Program EDITOR calls INIT and, using SWAP, fills arrays IAP and XTEMP. IAP contains the portion of the INDEX file contained in COMMON block CLOSE (the three closest neighbor stations for each station on the INDEX file); XTEMP contains

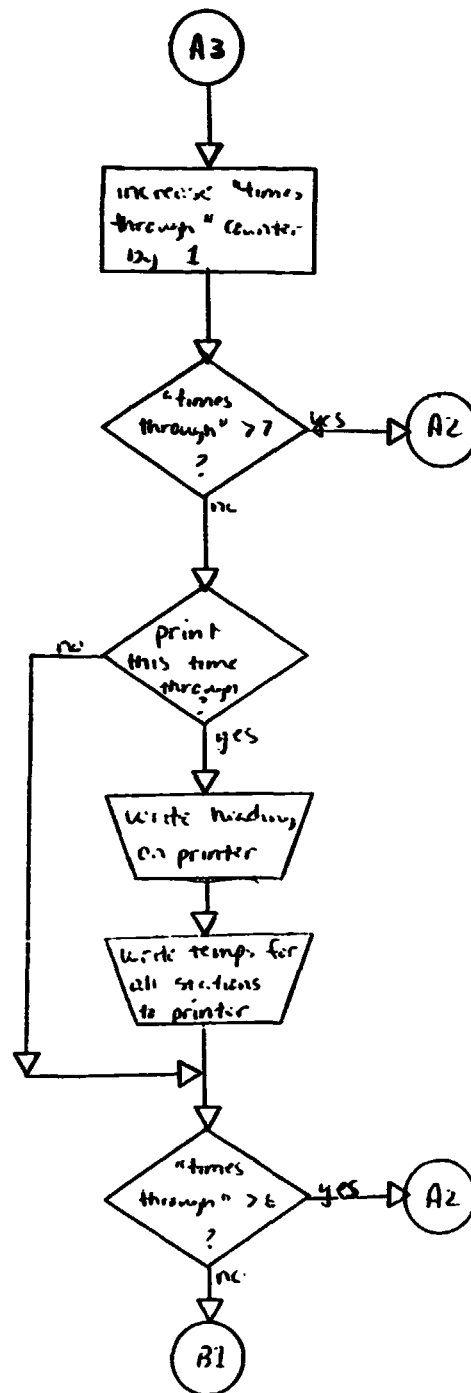
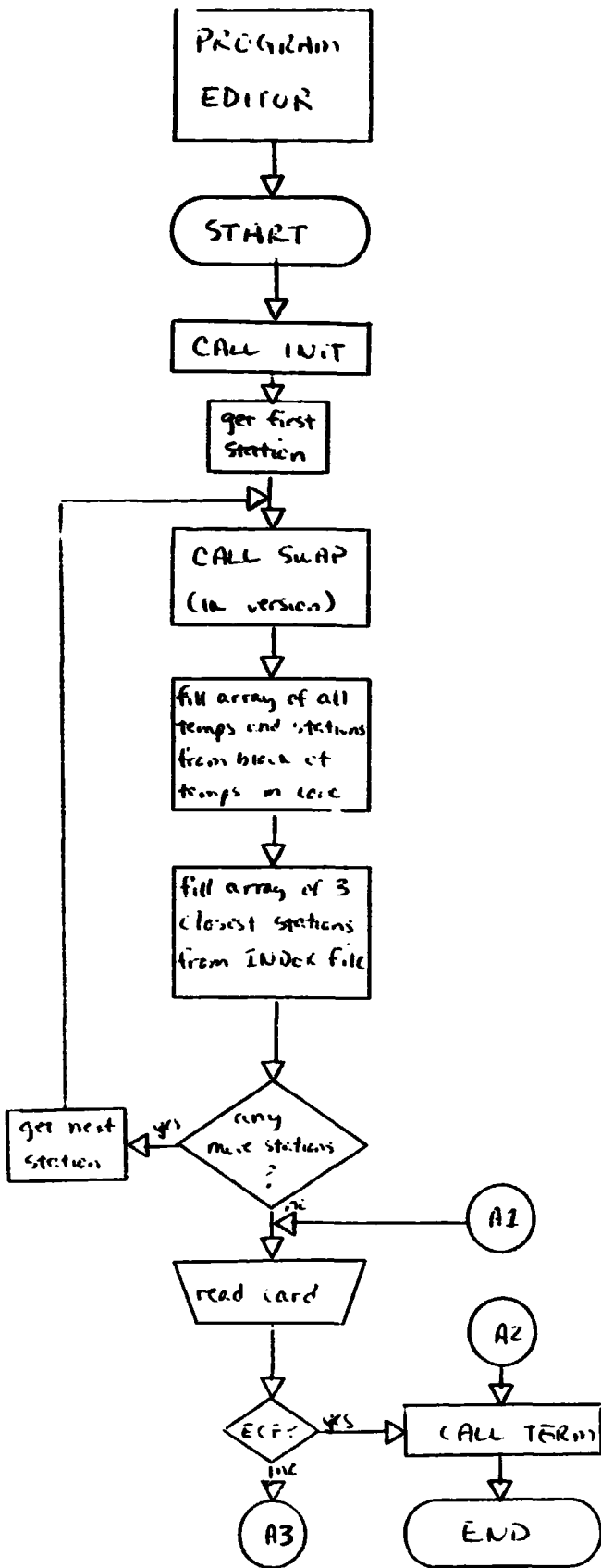
file CROPDATA. EDITOR then reads a card, obtaining values for variables IP, IC, IE, and IS. Each time a card is read, EDITOR increments the variable IL and checks for end-of-file. The edit process is executed three times. The first time, the contents of CROPDATA are listed; CHECK is called to identify values that need to be replaced; REPLAC is called and messages are written to the printer. The second execution of the edit process calls both CHECK and REPLAC. During the third pass through the program, CHECK is called and CROPDATA is rewritten from XTEMP. TERM is called after the completion of the third execution.

#### 3.2.12.6 Flowchart

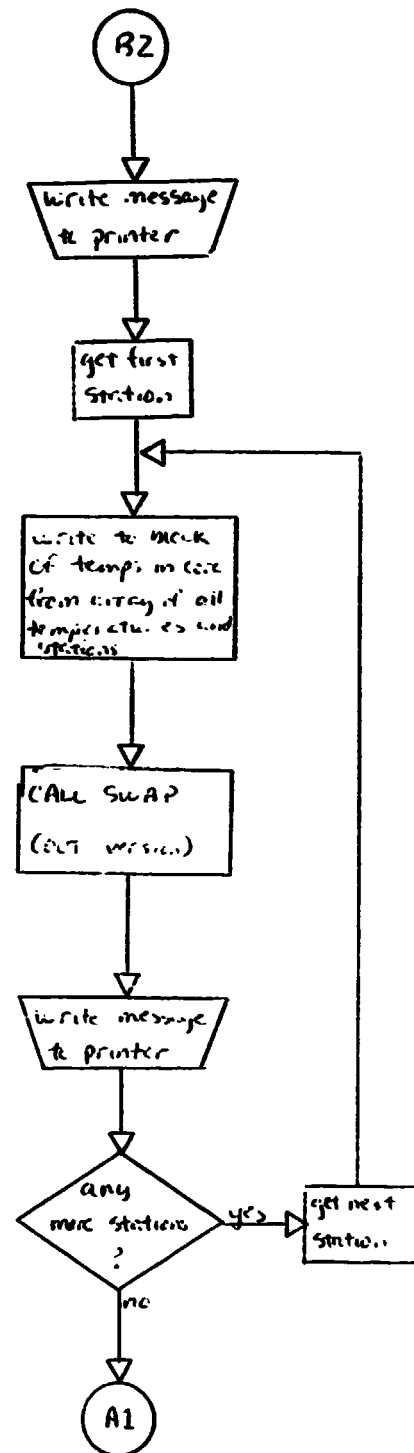
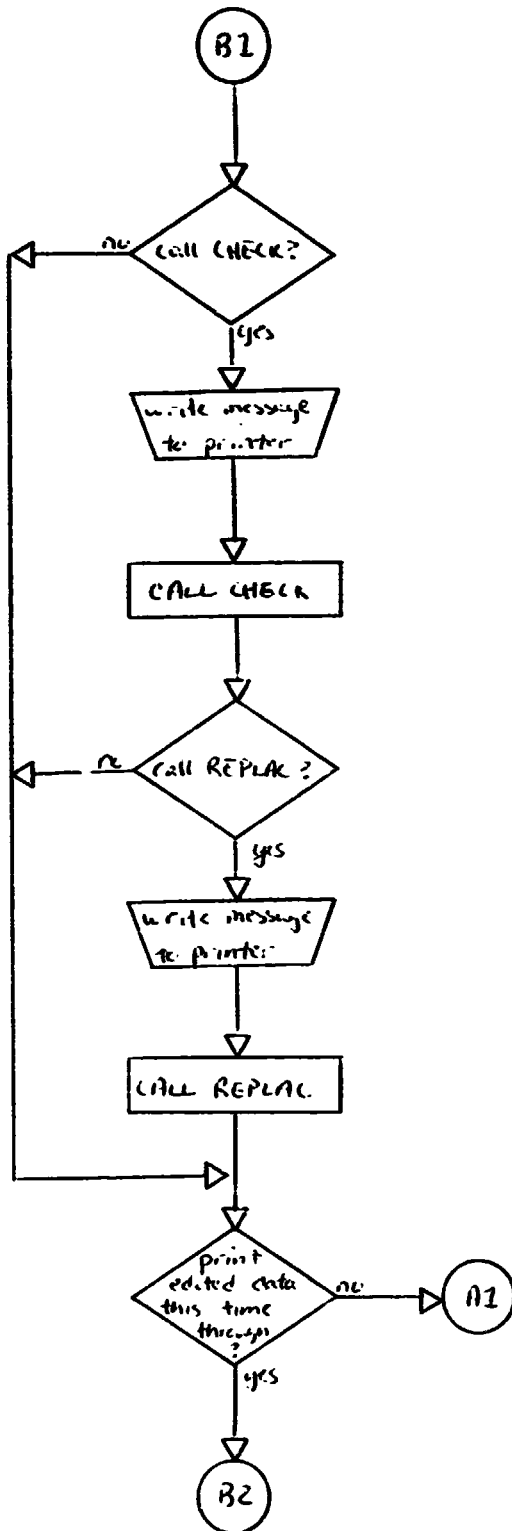
Next page.

#### 3.2.12.7 Listing

Follows flowchart.



# PROGRAM EDITOR, CONT



47

ORIGINAL PAGE IS  
OF POOR QUALITY

# PROGRAM EDITOR

## MAIN PROGRAM FOR CROP CALENDAR EDITING

```

REAL XTEMP(511,14,3),XMAP(511,3),XSD(511,3)
INTEGER LOW(511,3),HIGH(511,3),WMO(511),TEMPS(14,2),STATN(511,3)
INTEGER ITYPE(14),MIN(14),MAX(14),JF(14)
INTEGER IN/17,OUT/27,JULIAN,IP,IC,IE,IS,IL
INTEGER NT(511,3),IAP(511,3)
INTEGER*2 NUMSTA,FILLF
LOGICAL*1 TFLAG(511,15,2)
COMMON /INDEX/ NUMSTA,FILLF,WMO
COMMON /VAL1/ XTEMP,XMAP,XSD,LOW,HIGH,NT,IAP,JULIAN,TFLAG
COMMON /CLOSE/ STATN
CALL INIT
DO 50 K = 1,NUMSTA
CALL SWAP(IN,WMO(K),TEMPS,JULIAN)
DO 30 I = 1,3
IAP(K,I)=0
DO 10 J = 1,NUMSTA
IF (WMO(J).EQ.STATN(K,I)) GO TO 20
10 CONTINUE
GO TO 30
20 IAP(K,I)=J
30 CONTINUE
DO 40 J = 1,14
DO 40 I = 1,2
IF (TEMPS(J,I).EQ.999) TEMPS(J,I)=9999
XTEMP(K,J,I)=TEMPS(J,I)
40 CONTINUE
50 CONTINUE
II=0
20001 IF (II.GT.2000) II=9000 IP,IC,IE,IS
20001 FORMAT(2014)
II=II+1
IF (II.GT.7) GO TO 9000
IF (IP.NE.1) GO TO 100
WRITE(6,10001) II
10001 FORMAT(' CROP CALENDAR DATA LISTER -- PASS',I3)
DO 40 K = 1,NUMSTA
WRITE(6,10002) K
10002 FORMAT(' ')
DO 70 I = 1,2
J=3-I
WRITE(6,10003) WMO(K),ITYPE(J),JULIAN,(XTEMP(K,I,J),I=1,14)
10003 FORMAT('14.15.14.A4.I3.14F6.0')
70 CONTINUE
80 CONTINUE
100 IF (II.GT.6) GO TO 9000
IF (IC.LT.1) GO TO 300
WRITE(6,10004) II
10004 FORMAT(' CROP CALENDAR CHECKER -- PASS',I3)
CALL CHECK(IC,II)
IF (II.LT.1) GO TO 300
WRITE(6,10005) II
10005 FORMAT(' CROP CALENDAR EDITOR -- PASS',I3)
CALL REFLAC(II)
300 IF (IS.NE.1) GO TO 40
WRITE(6,10006) II
10006 FORMAT(' CROP CALENDAR REWRITE FILE -- PASS',I3)
DO 320 K = 1,NUMSTA
DO 310 I = 1,2
DO 310 J = 1,14
TEMPS(J,I)=XTEMP(K,J,I)+0.5
310 CONTINUE
320 CONTINUE

```

ORIGINAL PAGE IS  
OF POOR QUALITY



```
C      CALL SWAP(OUT,WMO(K),TEMP,JULIAN)
      WRITE(5,10007) WMO(K)
10007  FORMAT('0',I5,' REWRITTEN TO FILE')
      CONTINUE
      GOTO 60
60009  CALL TEMP
      STOP
      END
```

64

### 3.2.13 JCL FILE EDIT

Executing JCL file EDIT invokes the EDITOR program.

#### 3.2.13.1 Linkages

None.

#### 3.2.13.2 Interfaces

EDIT should not be executed until CAPTURE has successfully filled file CROPDATA.

#### 3.2.13.3 Inputs

Direct card input to this file is necessary. See Section 4.2.2.3 for further discussion and input format.

#### 3.2.13.4 Outputs

None.

#### 3.2.13.5 Description

PROC EDIT binds the files referenced by program EDITOR to that program and requests the invocation of EDITOR.

#### 3.2.13.6 Listing

```
//EDITPROC
//EDITEXECPGM=EDITOR,TIME=(,15)
//STEPLIBDDDSN=W.EDS.CCEA.PHASE3.LOAD,DISP=SHR
//FT02F001DDDSN=W.EDS.CCEA.{GROUPI}.INDEX,DISP=SHR
                        {GROUPII}
//FT04F001DDDSN=W.EDS.CCEA.{GROUPI}.CROPDATA,DISP=SHR
                        {GROUPII}
//FT05F001DDDUMMY
//FT06F001DDSYSOUT=A
//PPPPPEND
```

### 3.2.14 SUBROUTINE CHECK

CHECK determines which data in CROPDATA should be replaced.

#### 3.2.14.1 Linkages

CHECK uses COMMON block INDEX and VAL1.

#### 3.2.14.2 Interfaces

None.

#### 3.2.14.3 Inputs

IC = 1 do not print results;  
      = 2 print results  
IL the number of times through the edit procedure for  
      this station.  
XTEMP(I,14,3) an array containing the information on  
              file CROPDATA, as well as space for the  
              daily difference between maximum and  
              minimum temperature, where I is the  
              number of crop calendar stations.  
IAP(I,3) an array containing the positions in  
          array WMO of the three closest neighbor  
          stations for each crop calendar station,  
          where I is the number of stations.

#### 3.2.14.4 Outputs

Outputs from CHECK contained in COMMON block VAL1 and where  
I is the number of stations involved:

XBAR(I,3) mean of each station's max,min temperatures  
          and their differences.  
XSD(I,3) standard deviations of each station's three  
          temperature types.  
LOWB(I,3) lower confidence bounds of each station's  
          three temperature types.  
HIGHB(I,3) upper confidence bounds of each station's  
          three temperature types.  
NT(I,3) number of acceptable values of each station's  
          three temperature types.  
TFLAG(I,15,2) a logical array indicating "good" or "bad"  
              temperatures; the 15th position is flagged  
              "good" if all values for that temp. type are  
              "good" and "bad" if at least one temp. is "bad".

CDAY(14,2) array for printed output indicating values that should be manually checked and edited.

#### 3.2.14.5 Description

CHECK begins by counting reasonable temperatures and flagging unreasonable ones. When a station has at least 5 "good" max temps. and/or 5 "good" min temps. for the two-week period, the mean temperature, variance, standard deviation, and upper and lower confidence bounds of the temperature type are calculated. When both max and min are present for a day, differences between the two are calculated and summed; provided at least 5 differences for a station exist, the same statistics as for the max and/or min are calculated. CHECK attempts to find and calculate similar data for a station's three closest neighbor stations; 4-station averages of the mean temperature, standard deviation, lower and upper confidence bounds are figured. Whenever a large variation occurs between the mean of a temperature type and the corresponding 4-station mean, the 4-station mean is substituted.

Temperatures falling outside of the upper and lower confidence bounds are flagged. Then the maximum of one day is compared to the minimum of the next; whenever too much or no variation between the two are encountered, temperatures are flagged according to the type of variation. After all temperatures for a particular station have been checked, an array CDAY, denoting acceptable vs. not acceptable temperatures is filled. Beginning with the checks outside the upper and lower confidence bounds, the entire process is repeated. After the second pass through this part of CHECK, the value of IC is checked. For IC=1, no printout is generated. but for IC=2, the following is printed: a listing of max and min temperatures and their differences, the number of values of each, the mean, standard deviation, and lower and upper confidence bounds of each temperature, as well as the array CDAY indicating values that should be manually checked and edited before procedure SUZYQ is run.

#### 3.2.14.6 Flowchart

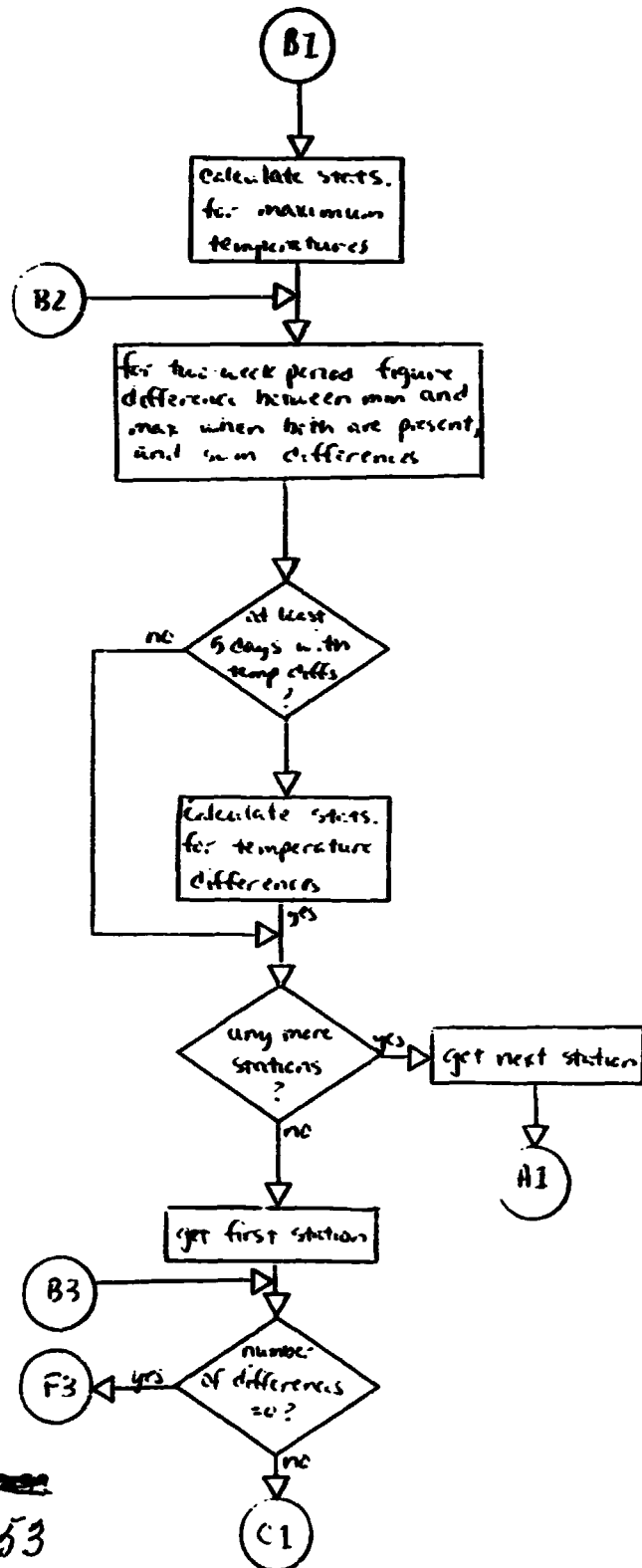
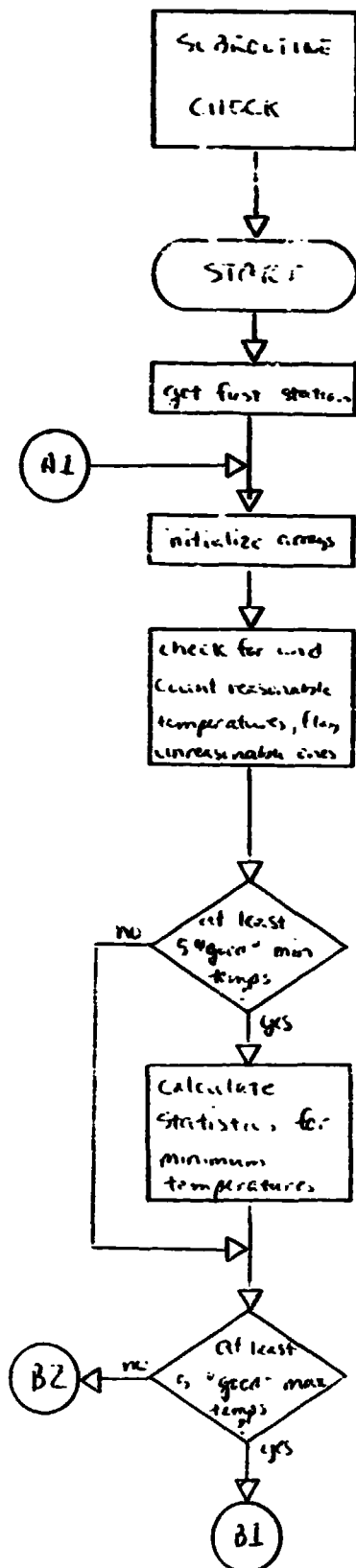
Next page.

#### 3.2.14.7 Listing

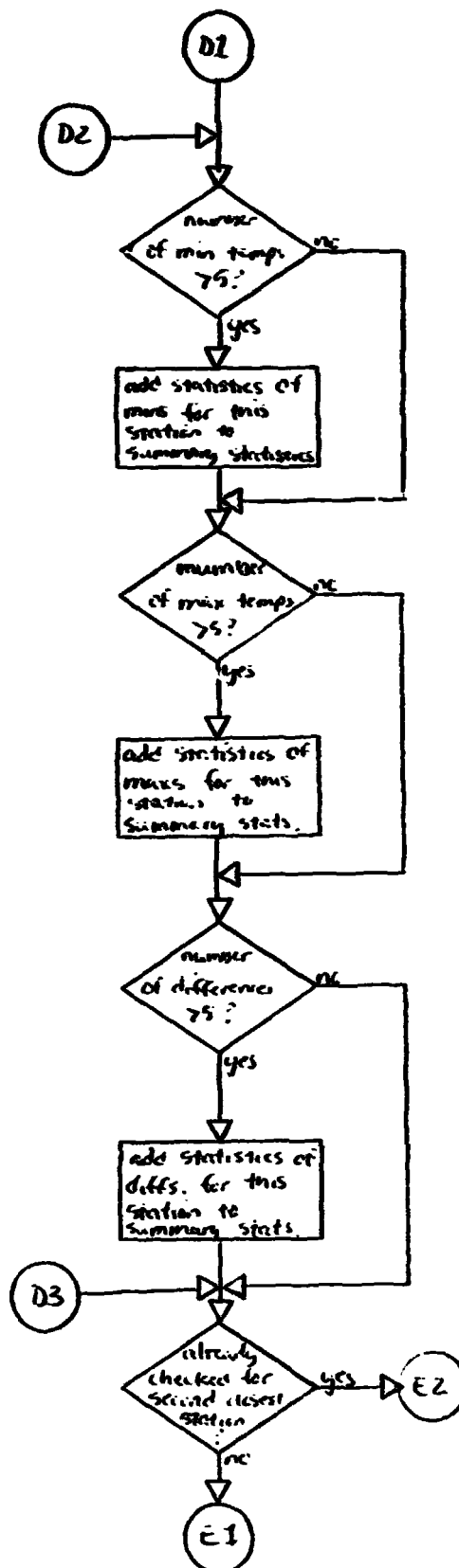
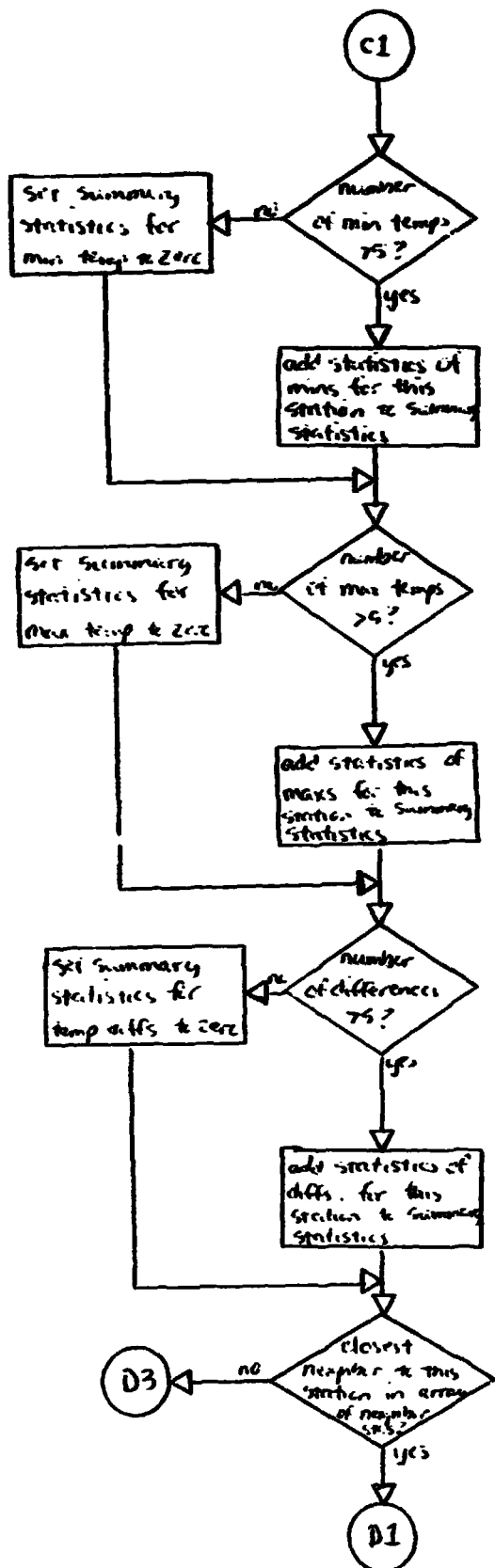
Follows flowchart.

~~3-90~~

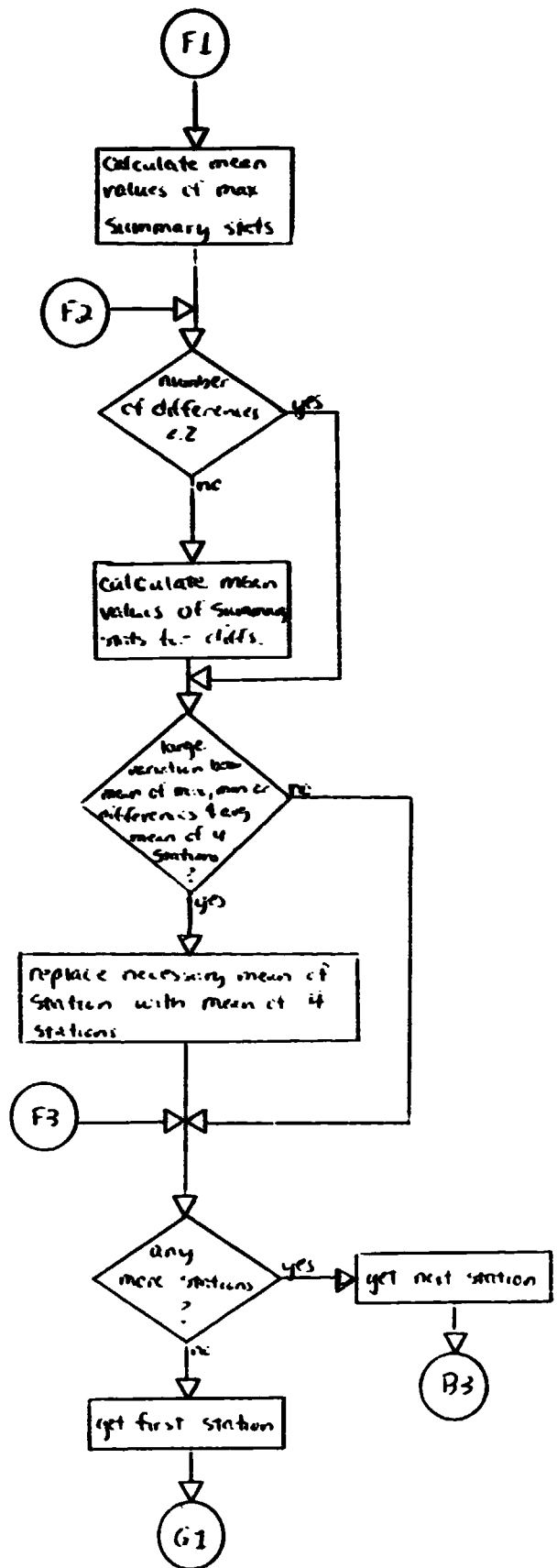
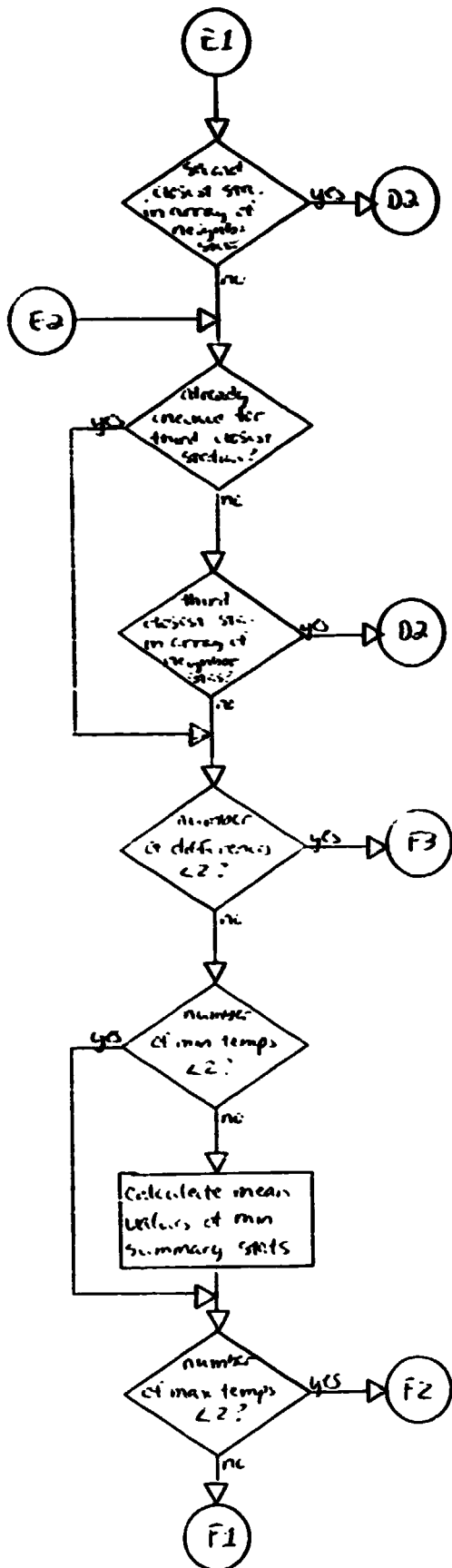
ORIGINAL PAGE IS  
OF POOR QUALITY



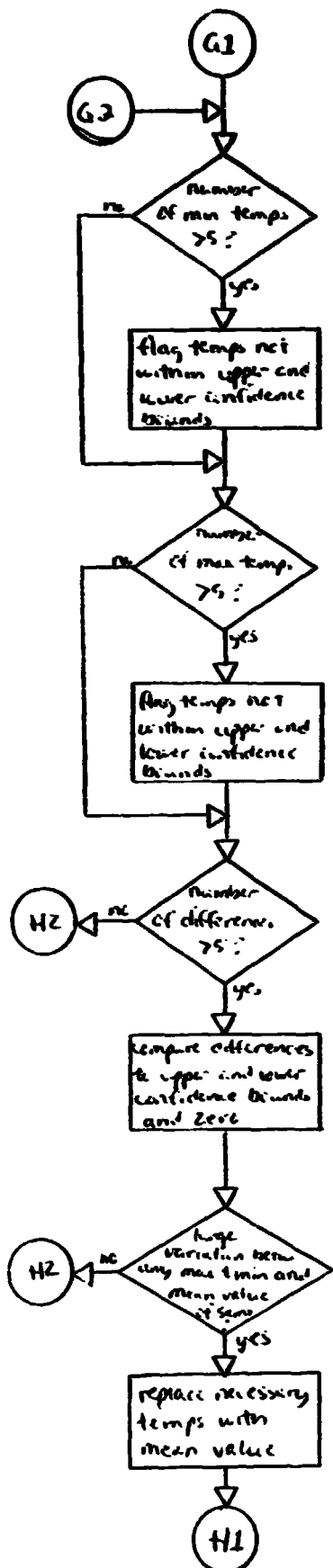
# SUBROUTINE CHECK, CONT



# SUBROUTINE CHECK, CON IT

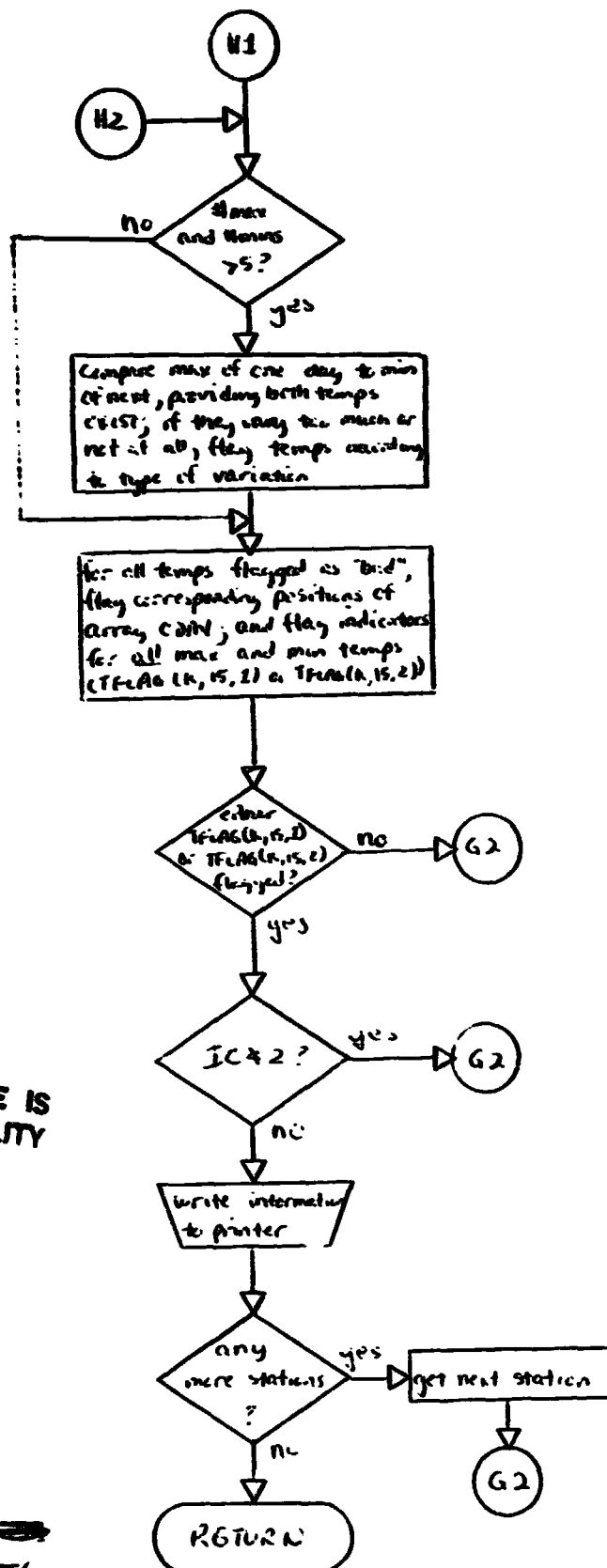


# SUBROUTINE CHECK, CONT



ORIGINAL PAGE IS  
OF POOR QUALITY

56





```

SUMPOUTLINE CHECK(IC,IL)
DEFAL XTE44(511.14.3).X44R(511.3).XSD(511.3).A44R(3).ASD(3).RN(3).
  SX(3)

```

```

DEFAL ZK1(4)/1.565.1.95.2.326.2.744.3.291.3.891.
DEFAL ZK2(4)/1.16.2.326.2.744.3.291.3.891.4.417.
DEFAL Z0.21.744.3.291.3.891.4.417.
INTEGER LOW(511.3).HIGH(511.3).MO(511).TEMPS(14.2).CDAY(14.2).
  A(3).LOWA(3).HIGA(3)
INTEGER ITYPE(3)/MIN 1.1MAX 1.1DIF 1.
INTEGER HIGH/110.1LOW/-50.1CXX/1 XX 1.1BLANK/1 1.1JULIAN.
  FULL.IL.1C

```

```

INTEGER NT(511.3).IAP(511.3)
INTEGER NT2(511.3).FILLER
LOGICAL TFLAG(511.15.2)
COMMON /INDEX/ NUMSTA.FILLER.WMO
COMMON /VAL1/ XTEMP.X44R.XSD.LOW.HIGH.NT.IAP.JULIAN.TFLAG
DO 70 K = 1.1NUMSTA
DO 10 I = 1.3
  NT(I)=0.0
  SX(I)=0.0
  X44R(K.1)=0.0
  XSD(K.1)=0.0
  NT(K.1)=0
  LOW(K.1)=0
  HIGH(K.1)=0

```

10 CONTINUE

```

DO 20 J = 1.2
  TFLAG(K.1.1)=.FALSE.
  DO 14 J = 1.14
    TFLAG(K.J.1)=.TRUE.
    TEMPS(J.1)=XTEMP(K.J.1)+0.5
    IF (TEMPS(J.1)-HIGH) 12.16.16
    IF (TEMPS(J.1)-LOW) 14.16.16
  14 DN(I)=RN(I)+1
  SX(I)=SX(I)+XTEMP(K.J.1)
  TFLAG(K.J.1)=.FALSE.

```

14 CONTINUE

20 CONTINUE

```

DO 52 I = 1.2
  IF (RN(I)-5.0) 54.52.52
  X44R(K.1)=SX(I)/DN(I)
  NT(K.1)=RN(I)
  X0=0.0
  X1=54 J = 1.14
  IF (TFLAG(K.J.1)) GOTO 54
  X2=XTEMP(K.J.1)-X44R(K.1)
  X0=X0+X2*X2

```

54 CONTINUE

```

XSD(K.1)=SQRT(XR/(RN(I)-1.0))
74=XSD(K.1)*K(11)
LOW(K.1)=X44R(K.1)-74
HIGH(K.1)=X44R(K.1)+74+0.5

```

55 CONTINUE

```

DO 62 J = 1.14
  IF (TFLAG(K.J.1).AND.(TFLAG(K.J.2))) GOTO 62
  DN(3)=DN(3)+1
  XTEMP(K.3)=TEMPS(J.2)-TEMPS(J.1)
  SX(3)=SX(3)+XTEMP(K.J.3)

```

62 CONTINUE

```

IF (DN(3)-5.0) 70.54.54
  X44R(K.3)=SX(3)/DN(3)
  NT(K.3)=DN(3)
  X0=0.0

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

60 DO J = 1,14
  IF ((IFLAG(K,J,1)).OR.(IFLAG(K,J,2))) GOTO 86
  X=X+TEMP(K,J,3)+HAR(K,J)
  Y=Y+XZPAZ
65 CONTINUE
  ZSD(K,3)=SQRT(42/(RN(3)-1.0))
  Z=ZSD(K,3)/ZSD(1,1)
  LON(K,3)=RN(3)-Z
  HIG(K,3)=KGA/(Z,3)+2470.5
70 CONTINUE
  DO I=1,N
    IF (I(K,3).EQ.0) GOTO 100
    Z78(I)=1.3
    IF (INT(K,1)-5) 74,72,72
72  IAT(I)=1
    AHA(I)=XHAZ(K,I)
    ASU(I)=ASU(K,I)
    LON(I)=LON(K,I)
    HIG(I)=HIG(K,I)
    GOTO 76
74  IAT(I)=0
    AHA(I)=0.0
    ASU(I)=0.0
    LON(I)=0
    HIG(I)=0
76  CONTINUE
  DO J=1,14
    IF ((I(K,J,1).LT.1) GOTO 86
    IAT(K,J)=1
    DO M=1,14
      IF (INT(J,M)-5) 44,42,62
42  NAT(I)=NAT(I,1)
    AHA(I,M)=XHAZ(I,4)+XHAZ(J,4)
    ASU(I,M)=ASU(I,1)+ASU(J,4)
    LON(I,M)=LON(I,1)+LON(J,4)
    HIG(I,M)=HIG(I,1)+HIG(J,4)
84 CONTINUE
86 CONTINUE
  IF (I(K,3).LT.2) GOTO 100
  DO I=1,N
    IF (I(K,1).EQ.2) GOTO 88
    AHA(I)=AHA(I)/ZFLON(IAT(I))
    ASU(I)=ASU(I)/ZFLON(IAT(I))
    LON(I)=LON(I)/IAT(I)
    HIG(I)=HIG(I)/IAT(I)
88 CONTINUE
  ZH=XHAZ(K,3)-AHA(I)
  IF (ZHAAS(I,3) .EQ. 100.100)
41  ZHAAS(AHA(K,2)-AHA(I,2))
    ZHAAS(ZHAZ(K,1)-AHA(I,1))
    I=I+10) 42,74,70
42  I=1
  GOTO 94
44  I=1
  Z=1
  GOTO 94
46  I=1
  Z=1
  GOTO 94
48  I=1
  Z=1
  GOTO 94
49  DO M=1,14
    AHA(K,M)=AHA(I,M)
    ASU(K,M)=ASU(I,M)
    LON(K,M)=LON(I,M)
    HIG(K,M)=HIG(I,M)
50 CONTINUE
    AHA(K,3)=AHA(I,3)
    ASU(K,3)=ASU(I,3)
    LON(K,3)=LON(I,3)

```

```

HIGHM(K,J)=HIGH(J)
100 CONTINUE
DO 200 K = 1,NUMSTA
DO 110 I = 1,2
IF (NT(K,I)-5) 110,102,102
102 DO 105 J = 1,14
IF (FLAG(K,J,1)) GOTO 108
T=TEMP(K,J,1)*0.5
IF (IT-LD(K,I)) 105,104,104
104 IF (IT-HIGHM(K,I)) 108,108,106
FLAG(K,J,1)=TRUE.
108 CONTINUE
110 CONTINUE
IF (NT(K,2)-5) 130,112,112
DO 125 J = 1,14
IF (TEMP(K,J,1).OR.FLAG(K,J,2)) GOTO 126
T=TEMP(K,J,1)
IF (IT-HIGHM(K,2)) 114,114,118
IF (IT-LD(K,2)) 114,114,118
IF (IT-LD(K,2)) 114,120,120
X=ABS(ATEMP(K,J,2)-XAS(K,2))
X=ABS(ATEMP(K,J,1)-XAS(K,1))
IF (IT-LD(K,2)) 120,122,124
121 FLAG(K,J,2)=TRUE.
GOTO 125
122 FLAG(K,J,2)=TRUE.
FLAG(K,J,1)=TRUE.
GOTO 120
124 FLAG(K,J,1)=TRUE.
125 CONTINUE
130 IF (NT(K,1).LT.5).OR.(NT(K,2).LT.5) GOTO 140
DO 135 J = 1,14
IF (FLAG(K,J,1).OR.FLAG(K,J,2)) GOTO 138
T=TEMP(K,J,1)*0.5
IF (TEMP(K,J,1).OR.TEMP(K,J,2)) 132,132,134
X=ABS(ATEMP(K,J,2)-XAS(K,2))
X=ABS(ATEMP(K,J,1)-XAS(K,1))
IF (IT-LD(K,2)) 134,136,137
136 FLAG(K,J,2)=TRUE.
GOTO 135
137 FLAG(K,J,1)=TRUE.
138 CONTINUE
140 DO 150 I = 1,2
DO 145 J = 1,14
CAY(J,1)=CAY(K
IF (NOT.FLAG(K,J,1)) GOTO 148
CAY(J,1)=CAY(K
FLAG(K,15,1)=TRUE.
144 CONTINUE
150 CONTINUE
IF (CAY(15,1).FLAG(K,15,1).AND.(NOT.FLAG(K,15,2))) GOTO 200
IF (CAY(15,2)) GOTO 200
WRITE(10,1001)
10012 C=241(100)
DO 155 I = 1,2
J=1
DO 160 K=10013, X40(K), I, TYPE(J), JULIAN, (ATEMP(K,IM,1),IM=1,14),
X40(K), X40(K), X50(K,1), LOGH(K,1), HIGHM(K,1),
CAY(15,1), CAY(15,2), 14
10015 WRITE(10,10014) X40(K), I, TYPE(J), JULIAN, (ATEMP(K,IM,1),IM=1,14),
X40(K), X40(K), X50(K,1), LOGH(K,1), HIGHM(K,1),
CAY(15,1), CAY(15,2), 14
10014 CONTINUE

```

ORIGINAL PAGE IS  
OF POOR QUALITY



10015 FORMAT(1X,16,1X,A4,13,14F6.0,15,2F6.2,215)  
200 CONTINUE  
RETURN  
END

09

.....

### 3.2.15 SUBROUTINE REPLAC

REPLAC attempts to find and insert replacement values for temperatures flagged as unacceptable by CHECK.

#### 3.2.15.1 Linkages

REPLAC uses COMMON blocks INDEX and VAL1.

#### 3.2.15.2 Interfaces

None.

#### 3.2.15.3 Inputs

IE = 1 edit but do not print results

= 2 edit and print results

JULIAN ending date of the previous 2-week period

where I is the number of stations involved,

XTEMP(I,14,3)

IAP(I,3)

XBAR(I,3)

XSD(I,3)

LOWB(I,3)

HIGHB(I,3)

NT(I,3)

TFLAG(I,15,2)

See sections 3.2.14.3 and 3.2.14.4 for descriptions of these variables.

#### 3.2.15.4 Outputs

Replacement values inserted into array XTEMP (I,14,3), where I is the number of stations, as well as messages indicating a successful substitution or the inability to make a substitution.

#### 3.2.15.5 Description

This subroutine is executed twice each time it is called. REPLAC determines if any values for a particular station need to be replaced. When substitutions need to be made, several algorithms are available:

A) At least seven temperatures of the station and type in

~~3-15~~

- question are available, as well as data from alternate stations;
- B) at least seven temps are available, but no alternate station data;
  - C) fewer than seven temps are available.

Algorithm A will utilize up to four days of data from the primary station and up to five days of data from each alternate station, provided the stations also meet the criterion of at least seven "good" temperatures. For each station involved, the temperatures are standardized and averaged. A weighting factor is also calculated for each station, as well as an average weighting factor. The replacement value is the sum of the standardized value for the primary station and the average weighting factor.

Algorithm B will first attempt to find a replacement value using the other temperature of the day if it is available (i.e., the max to find the min or vice versa). If this is not successful, REPLAC will try to use the average of up to four days of data for the replacement temperature. If no alternate days are available within a prescribed vicinity of the day in question, the replacement value will be the mean max or min value of the entire 14-day period.

Algorithm C will use the average temperature of the alternate stations for the day in question as the replacement. If Algorithm C does not generate a replacement value, a message to that effect will be printed.

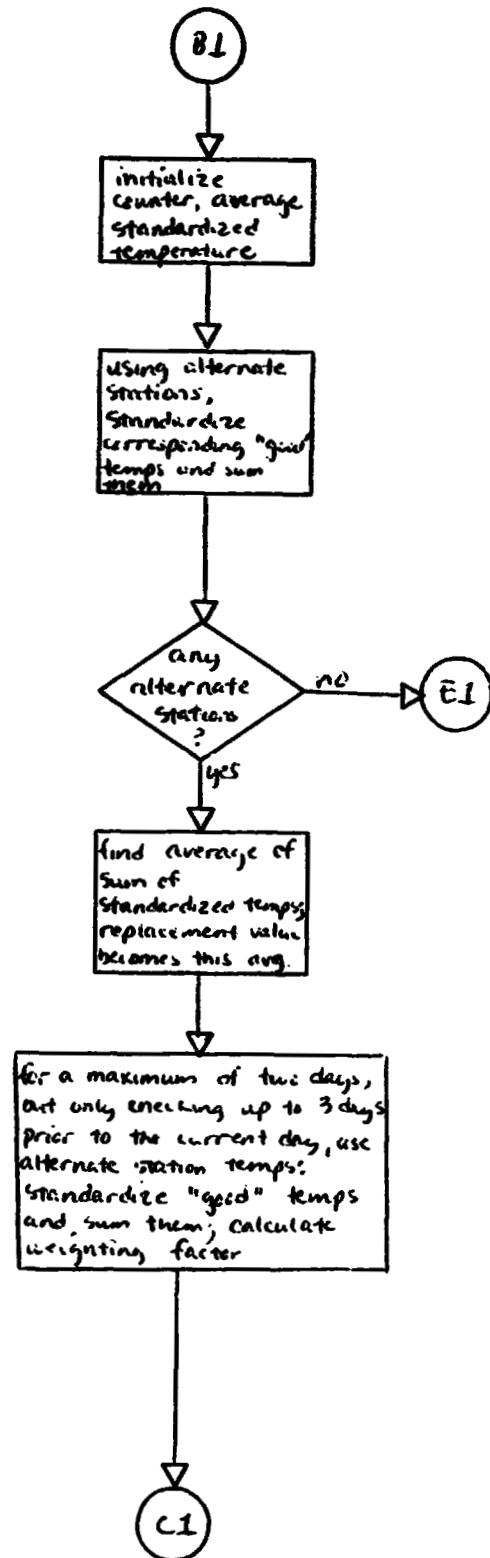
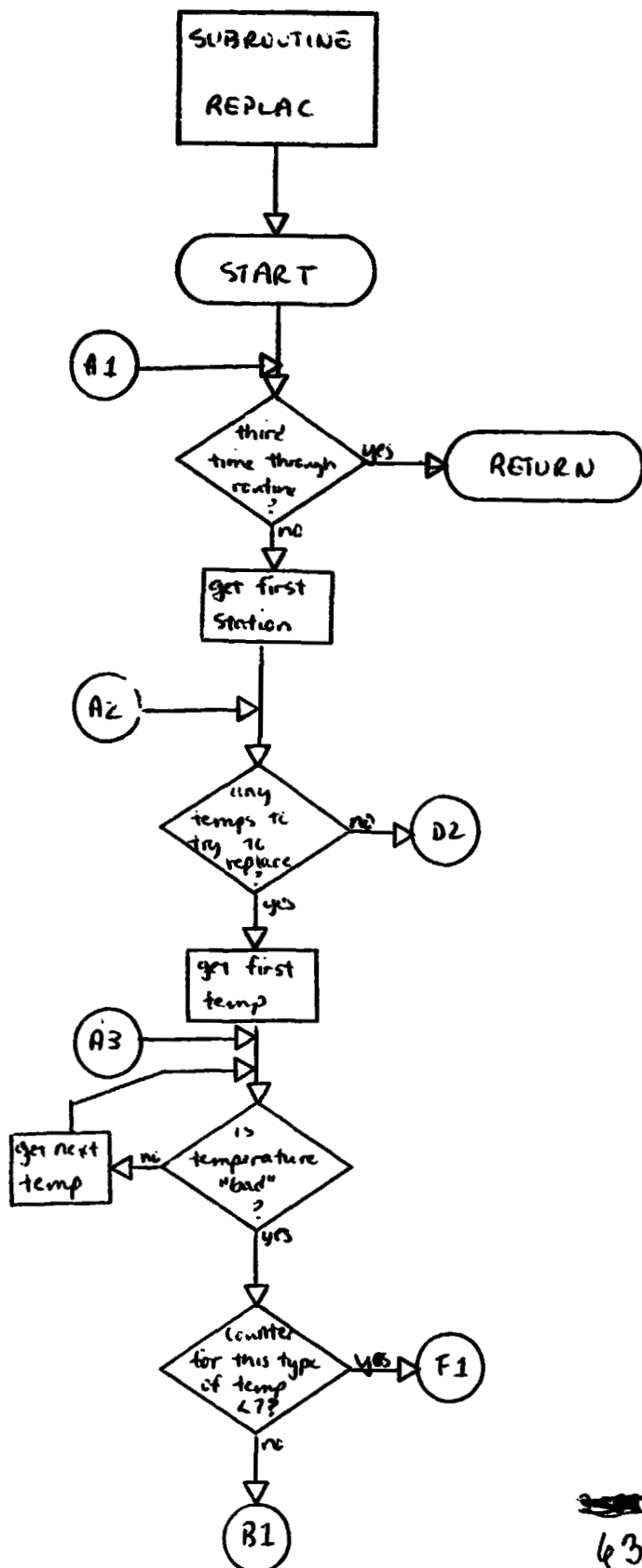
After substitutions for all stations needing them have been made and the subroutine has been executed twice, control returns to program EDITOR.

#### 3.2.15.6 Flowchart

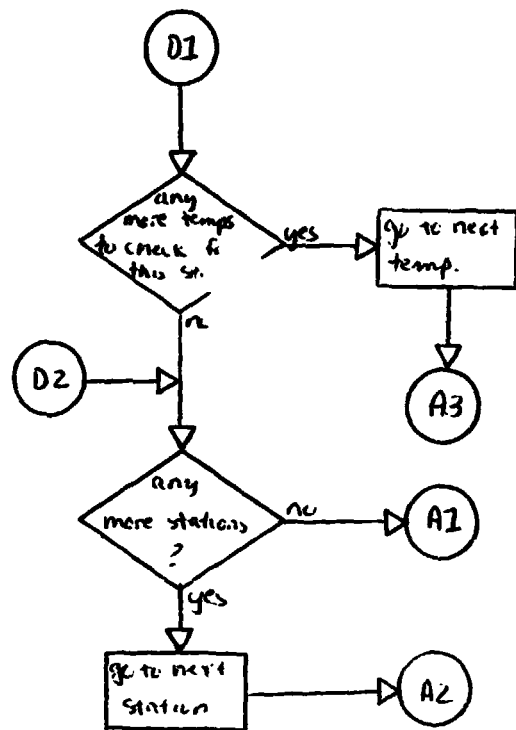
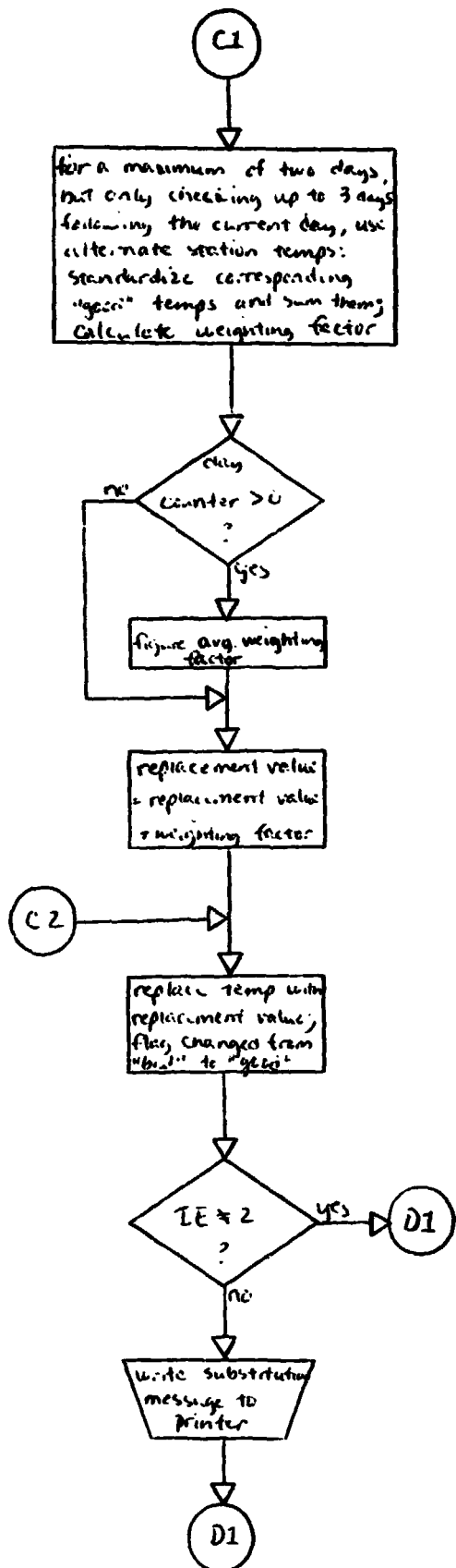
Next page.

#### 3.2.15.7 Listing

Follows flowchart.

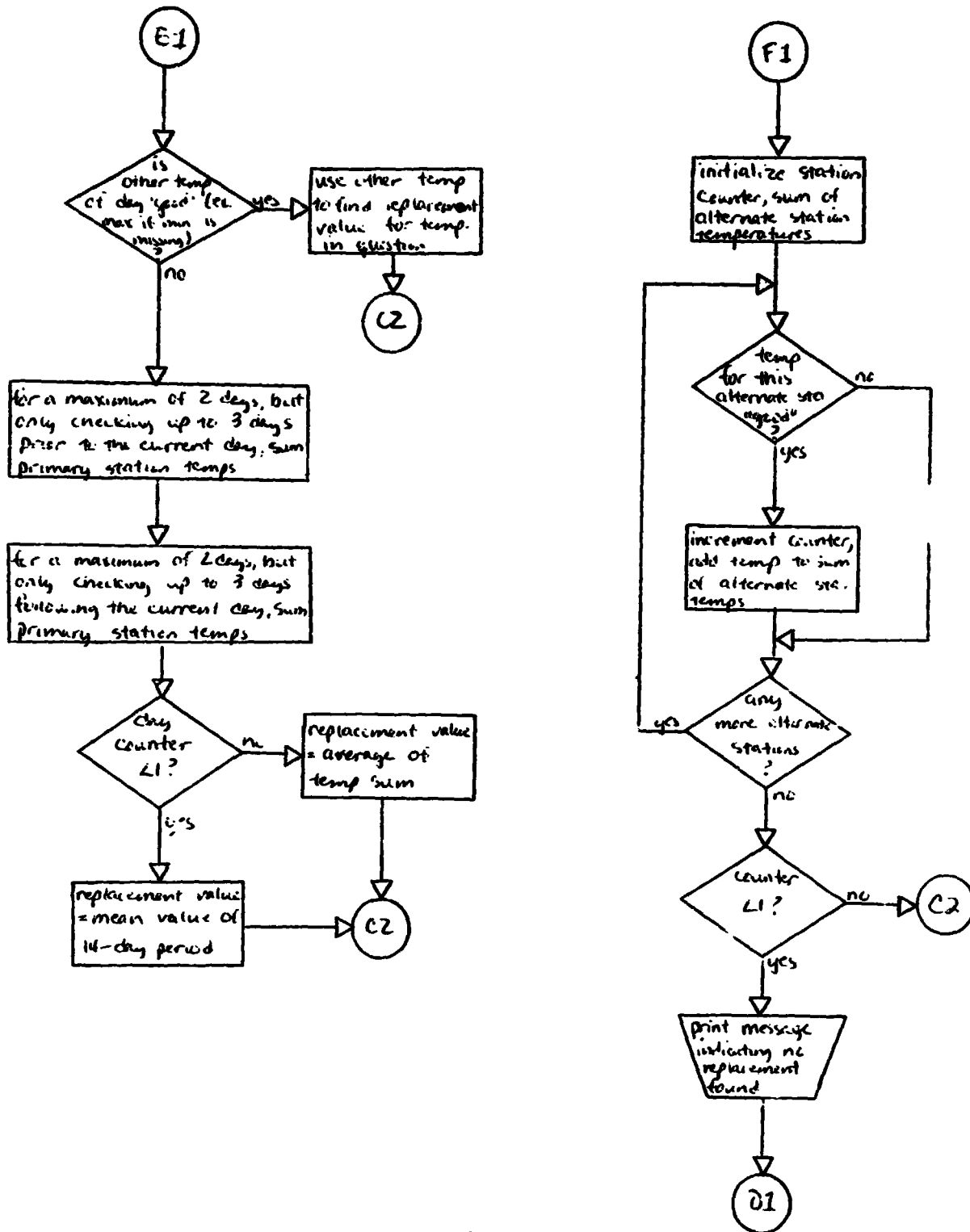


# SUBROUTINE REPLAC, CON.T





# SUBROUTINE REPLAC, CONT



SUBJECT

REPLACE MISSING VALUE USING MEANS AND STANDARD DEVIATIONS  
CHECK FOR ALTERNATE STATIONS

```

2040 I = 1.3
IF (IAP(ISTA,I).LT.1) GOTO 2040
4=IAP(ISTA,I)
I = IAP(R,ICAY,IM) GOTO 2040
IF (INTC(IM).LT.7) GOTO 2040

```

ALL DATE STATIONS AVAILABLE

CHECK FOR ALTERMATE STATION RIAS

7-1211.0  
\*721

71=0.0  
41=0

```
IF (FLAG(R,JDAY,M)) GOTO 2060
N1=N1+1
```

```
IF (N1.LT.1) GOTO 2065
X2=Z1/Z1*GAT(N1)*XSOLIS
```

$$72 = 75 \cdot (\text{ATE}4011572, \text{JDAY}, 1) - x(2)$$

```
IF (N2,LT,2) GOTO 2050
```

JDAY=JDAY+1

ORIGINAL PAGE IS  
OF POOR QUALITY

IF ((JDAY.GT.14).OR.(JDAY.GT.(IDAY+3))) GOTO 2100  
IF ((FLAG(ISTA,JDAY,IM)) GOTO 2080

Z1=0.0  
N1=0

DO 2080 I = 1,N

K=VAL(I)

IF ((FLAG(K,JDAY,IM)) GOTO 2090

Z1=Z1+(ATEMP(K,JDAY,IM)-XBAR(K,IM))/XSD(K,IM)

N1=N1+1

2090 CONTINUE

IF (N1.LT.1) GOTO 2095

Z2=Z1/FLOAT(N1)\*XSD(ISTA,IM)\*XBAR(ISTA,IM)

Z3=Z2+ATEMP(ISTA,JDAY,IM)-XZ)

N2=N2+1

2095 IF (N2.LT.4) GOTO 2080

2100 IF (N2.GT.0) Z4=Z2/FLOAT(N2)

Z5=XK-Z4

GOTO 2500

CCC  
\*\* NO ALTERNATE STATIONS AVAILABLE  
CHECK FOR OTHER TEMP WITHIN PRIMARY STATION

2110 I0=1

IF (IM.LT.1) I0=2

IF ((I0(ISTA,I0).LT.5) GOTO 2120

IF ((FLAG(ISTA,I0,IM)) GOTO 2120

Z3=ATEMP(ISTA,I0,IM)-XBAR(ISTA,I0)/XSD(ISTA,I0)

Z3=Z3+XSD(ISTA,IM)\*XBAR(ISTA,IM)

GOTO 2500

CCC  
\*\* NOT OTHER TEMP AVAILABLE  
CHECK SAME TEMPS WITHIN PRIMARY STATION

2120 IDAY=IDAY

Z4=0.0

N7=0

2130 IDAY=JDAY-1

IF ((JDAY.LT.1).OR.(JDAY.LT.(IDAY+3))) GOTO 2135

IF ((FLAG(ISTA,JDAY,IM)) GOTO 2130

Z4=Z4+ATEMP(ISTA,JDAY,IM)

N7=N7+1

IF (N7.LT.2) GOTO 2130

2135 IDAY=IDAY

2140 IDAY=JDAY-1

IF ((JDAY.GT.14).OR.(JDAY.GT.(IDAY+3))) GOTO 2145

IF ((FLAG(ISTA,JDAY,IM)) GOTO 2140

Z4=Z4+ATEMP(ISTA,JDAY,IM)

N7=N7+1

IF (N7.LT.4) GOTO 2140

2145 IF (N7.LT.1) GOTO 2150

Z5=Z4/FLOAT(N7)

GOTO 2500

CCC  
\*\* NO TEMP WITHIN PRIMARY STATION USEABLE  
USE PRIMARY STATION MEAN

2150 XBAR=Z1(ISTA,IM)

GOTO 2500

CCC  
\*\*\* NO STATION MEANS AND STANDARD DEVIATIONS  
CHECK FOR ALTERNATE STATIONS

2200 Z0=0.0

N0=0

DO 2210 I = 1,3

IF ((IAP(ISTA,I).LT.1) GOTO 2210

K=IAP(ISTA,I)

IF ((FLAG(K,IDAY,IM)) GOTO 2210

ORIGINAL PAGE IS  
OF POOR QUALITY



```

N=N+1
Z0=Z0+XTEMP(K,IDAY,IM)
2210 CONTINUE
IF (N.LT.1) GOTO 2600
C ** ALTERNATE STATION AVAILABLE
C X0=Z0/FLOAT(N)

```

```

C ***** REPLACE MISSING VALUE WITH SUBSITUTE VALUE
C 2500 XTEMP(ISTA,IDAY,IM)=X0
C IFLAG(ISTA,IDAY,IM)=.FALSE.
C IF (IE.NE.2) GOTO 3000
C JULI=JULIAN-IDAY-1
C WRITE (6,10021) #MO(ISTA),JULI,ITYPE(IM),X0
10021 FORMAT('STATION',I6,' JULIAN DAY',I4,IX,A4,
- XTEMP WAS MISSING. A VALUE OF',F5.0,' HAS BEEN SUBSITUTED.')
C GOTO 3000
C ***** NO VALUE FOUND FOR SUBSTITUTION
C 2600 JULI=JULIAN-IDAY-1
C WRITE (6,10022) #MO(ISTA),JULI,ITYPE(IM)
10022 FORMAT('STATION',I6,' JULIAN DAY',I4,IX,A4,
- XTEMP WAS MISSING. NO VALUE AVAILABLE FOR SUBSTITUTION. ****)
C 3000 CONTINUE
C 4000 CONTINUE
C 5000 CONTINUE
C RETURN
C END

```

### 3.2.16 PROGRAM CROPCALN

CROPCALN reads the weather data and models the crop progress.

#### 3.2.16.1 Linkages

CROPCALN calls subroutines INIT, SWAP, START, PHENO and TERM.

#### 3.2.16.2 Interfaces

CROPCALN is run after file CROPDATA has been edited.

#### 3.2.16.3 Inputs

The old master file is assumed on Unit 1, INDEX on Unit 2, and CROPDATA on Unit 4.

#### 3.2.16.4 Outputs

Unit 6 is the printer, GRIDINFO is on Unit 8, the new master file is on Unit 9, and ARNO is on Unit 11.

#### 3.2.16.5 Description

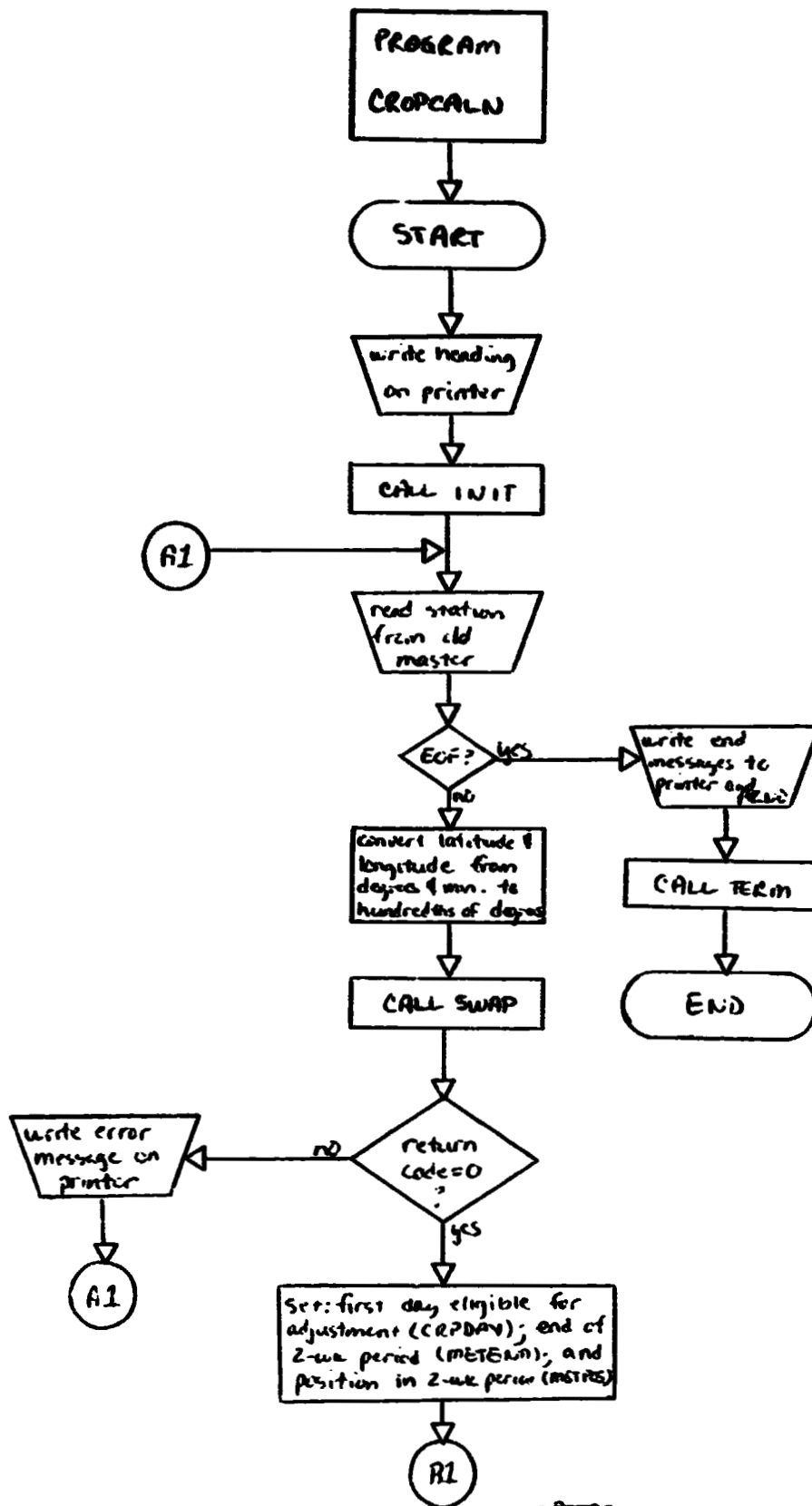
CROPCALN begins by writing a header to the printer and calling INIT. The program then enters a loop and reads a station. SWAP is called to obtain weather data from CROPDATA; CROPCALN checks and adjusts for a two-week period that spans two years and verifies the data obtained from CROPDATA, writing error messages ceasing operations for that particular station when errors are encountered. If the crop stage has not yet reached 1.0, START is called. If STAGE is between 1.0 and 6.0, CROPCALN goes into an inner loop; the daily crop progress for the two-week period is calculated via the subroutine PHENO and written to file GRIDINFO. Information is also written to file ARNO and the new master file. After all stations have been processed, ending messages are written to ARNO, the new master file, and the printer. TERM is called to print messages and close CROPDATA.

#### 3.2.16.6 Flowchart

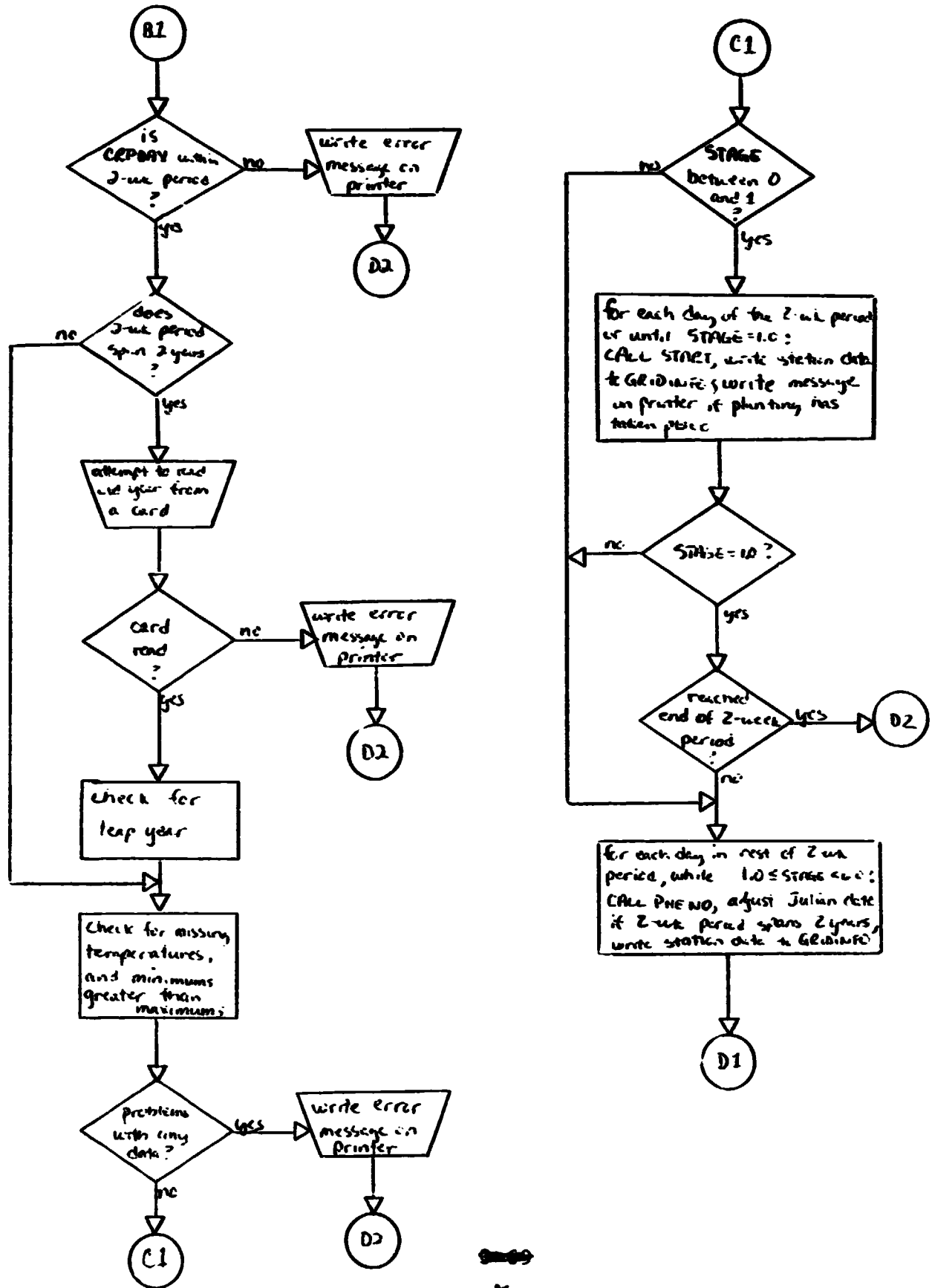
Next page.

#### 3.2.16.7 Listing

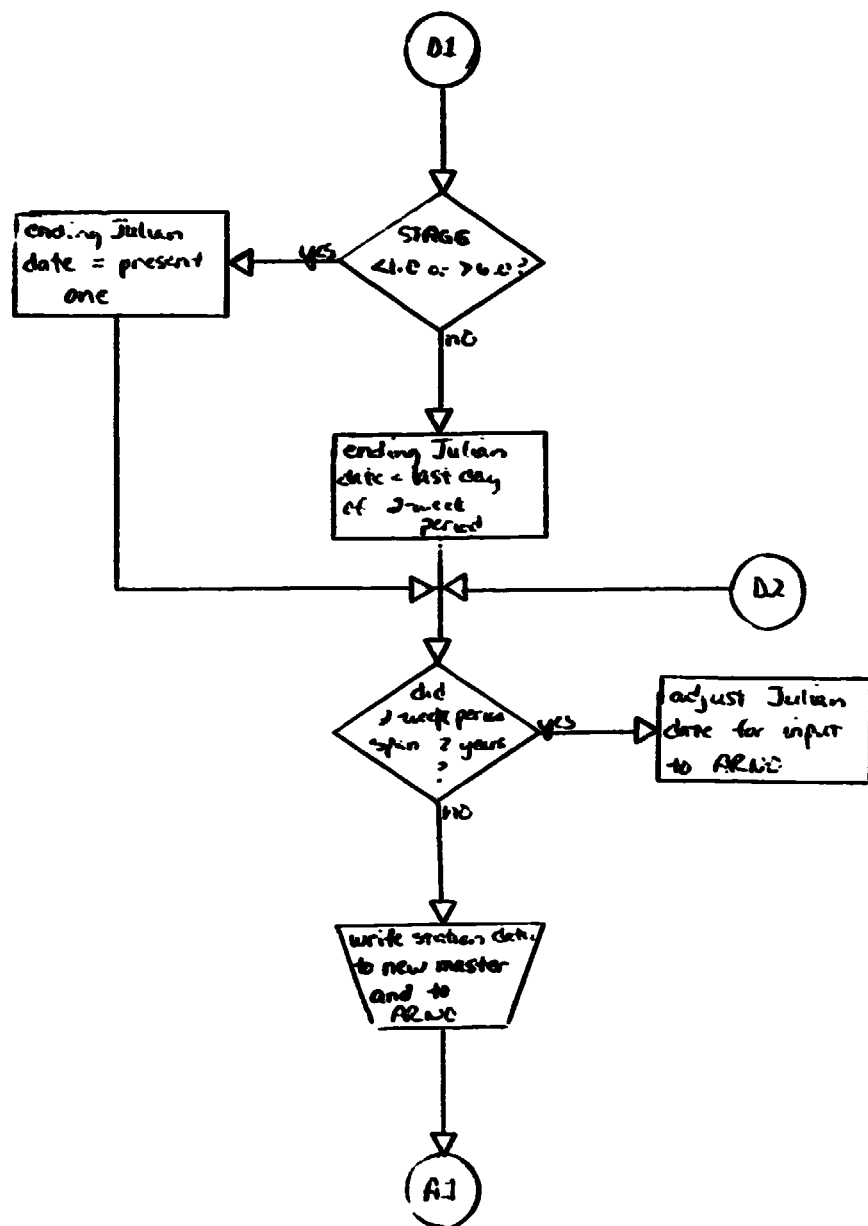
Follows flowchart.



# PROGRAM CRUCALAN, CONT



# PROGRAM (CHUPCHKA), CENT





3.2.16.7 L  
PROGRAM

ORIGINAL	PAGE	IS
DE POOR	QUALITY	

COB

MIN PHENOLOGICAL MODEL, IF APPROPRIATE

50 NOW = TODAY  
IF (TODAY.GT.METEND) GO TO 200  
DO GO TODAY = NOW + METEND  
IF (ISTAGE.LT.1.0).OR.(ISTAGE.GT.2.0) GO TO 70  
CALL PHENO(TODAY,STAGE,COOR,TEMPS(METROS.1),TEMPS(METROS.2),MULT.

ALAT,ALONG)  
TODAY=TODAY  
IF (TODAY.GT.365) TODAY=(365+K)  
IF (TODAY.GT.365+360) TODAY=1  
WRITE(4,1001)COOR,REGION,ZONE,STATA,ALAT,ALONG,COEV,CRNP,TODAY,  
STAGE,MULT,TEMPS(METROS.1),TEMPS(METROS.2)  
60 METROS = METROS + 1  
MULT = MULT + 1  
GO TO 200  
70 MULT = TODAY  
GO TO 200

C PHEN ROUTINES

C  
100 WRITE(4,1001)J,MND  
1001 FORMAT(1 MISSING DATA :.214:. STATION:1A)  
GO TO 200

C  
101 WRITE(4,1011)J,MND  
1011 FORMAT(1 INPUTED TEMPS :.214:. STATION:1A)  
GO TO 200

C  
102 WRITE(4,1021)METEND,COVJAY,MND  
1021 FORMAT(1 MET DATA FOR PAST PERIOD :.214:. STATION:1A)  
GO TO 200

C  
103 WRITE(4,1031)COVJAY,METEND,MND  
1031 FORMAT(1 MET DATA FOR FUTURE PERIOD :.214:. STATION:1A)  
GO TO 200

C  
104 WRITE(4,1041)  
1041 FORMAT(1 ERROR: TWO YEARS SPANNE!! FIRST YEAR MUST BE INPUT !!)

C  
200 LINE=JULIAN  
IF (JUL.GT.365) JUL=JUL-(365+K)  
IF (JUL.GT.365+360) JUL=1  
WRITE(5,STATA,COOR,REGION,ZONE,STATA,ALAT,ALONG,COEV,CRNP,1JUL,  
STAGE,MULT  
LINE = LINE + 1  
IF (LINE.EQ.50) GO TO 210  
DATE = PHEN

WRITE(11,2001) DATE  
2001 FORMAT(11,2001)CND CALENDAR SUMMARY PRINTOUT:15:1PAGE:14:7:10  
STATION=COOR-REGION-ZONE-STATA-ALAT-LONG-DEVELOPMENT-CRNP-J  
LY-STAGE-7)

210 LINE = 1  
WRITE(11,2101)MND,COV,REGION,ZONE,STATA,ALAT,ALONG,COEV,CRNP,  
1JUL,STAGE  
2101 FORMAT(17,31:2F12.2,F3.2,F14.1,F10.1)  
GO TO 1

C  
220 WRITE(4,2201)J,MND  
2201 FORMAT(1 STATION:15:1 NOT FOUND. IGNORED.)  
GO TO 1

C  
300 WRITE(11,3001)  
3001 FORMAT(11,3001)CND CALENDAR SUMMARY  
WRITE(4,3002)  
3002 FORMAT(11,3002)CND OF REPORT  
CALL TERM



175

ORIGINAL PAGE IS  
OF POOR QUALITY



### 3.2.17 JCL FILE SUZYQ

Executing JCL file SUZYQ invokes the crop calendar model.

#### 3.2.17.1 Linkages

None.

#### 3.2.17.2 Interfaces

None.

#### 3.2.17.3 Inputs

None.

#### 3.2.17.4 Outputs

None.

#### 3.2.17.5 Description

JCL file SUZYQ binds the files referenced by CROPCALN to that program and request the invocation of CROPCALN.

#### 3.2.17.6 Listing

```
//SUZYQ//PROC//OLD=,NEW=,DSP=MOD
//TRACK//EXEC//PGM=CROPCALN
//STEPLIB//DD//DSN=W.EDS.CCEA.PHASE3.LOAD,DISP=SHR
//FT01F001//DD//DSN=W.EDS.CCEA.{GROUP I}.MASTER&OLD.,DISP=SHR,
                                {GROUP II}
//FT02F001//DD//DSN=W.EDS.CCEA.{GROUP I}.INDEX,DISP=SHR,DCB=BUFNO=1
                                {GROUP II}
//FT04F001//DD//DSN=W.EDS.CCEA.{GROUP I}.CROPDATA,DISP=SHR,
                                {GROUP II}
//FT06F001//DD//SYSOUT=A
//FT08F001//DD//DSN=W.EDS.CCEA.{GROUP I}.GRIDINFO,DISP=&DSP.,
                                {GROUP II}
//FT09F001//DD//DSN=W.EDS.CCEA.{GROUP I}.MASTER&NEW.,DISP=SHR,
                                {GROUP II}
//FT11F001//DD//DSN=W.EDS.CCEA.{GROUP I}.ARNO,DISP=SHR,DCB=BUFNO=1
                                {GROUP II}
```

```

//REPORTEXECPGM=IEBTPCH
//SYSUT1DDDSN=W.EDS.CCEA.{GROUP I } .ARNO,DISP=SHR
                        {GROUP II }
//SYSINDDDSN=W.EDS.CCEA.DATALIB(PREFORM),DISP=SHR
//SYSPRINTDDDUMMY
//SYSUT2DDSYSOUT=A
//DDDDPEND

```

### 3.2.18 SUBROUTINE START

START accumulates temperature data to determine the planting date for individual spring wheat crop calendar stations.

#### 3.2.18.1 Linkages

None.

#### 3.2.18.2 Interfaces

None.

#### 3.2.18.3 Inputs and Outputs

CDEV	cumulative development variable
STAGE	value of Robertson Scale
CROP	winter wheat = 11, spring wheat = 12
MIN	minimum temperature for a given day
MAX	maximum temperature for a given day

#### 3.2.18.4 Description

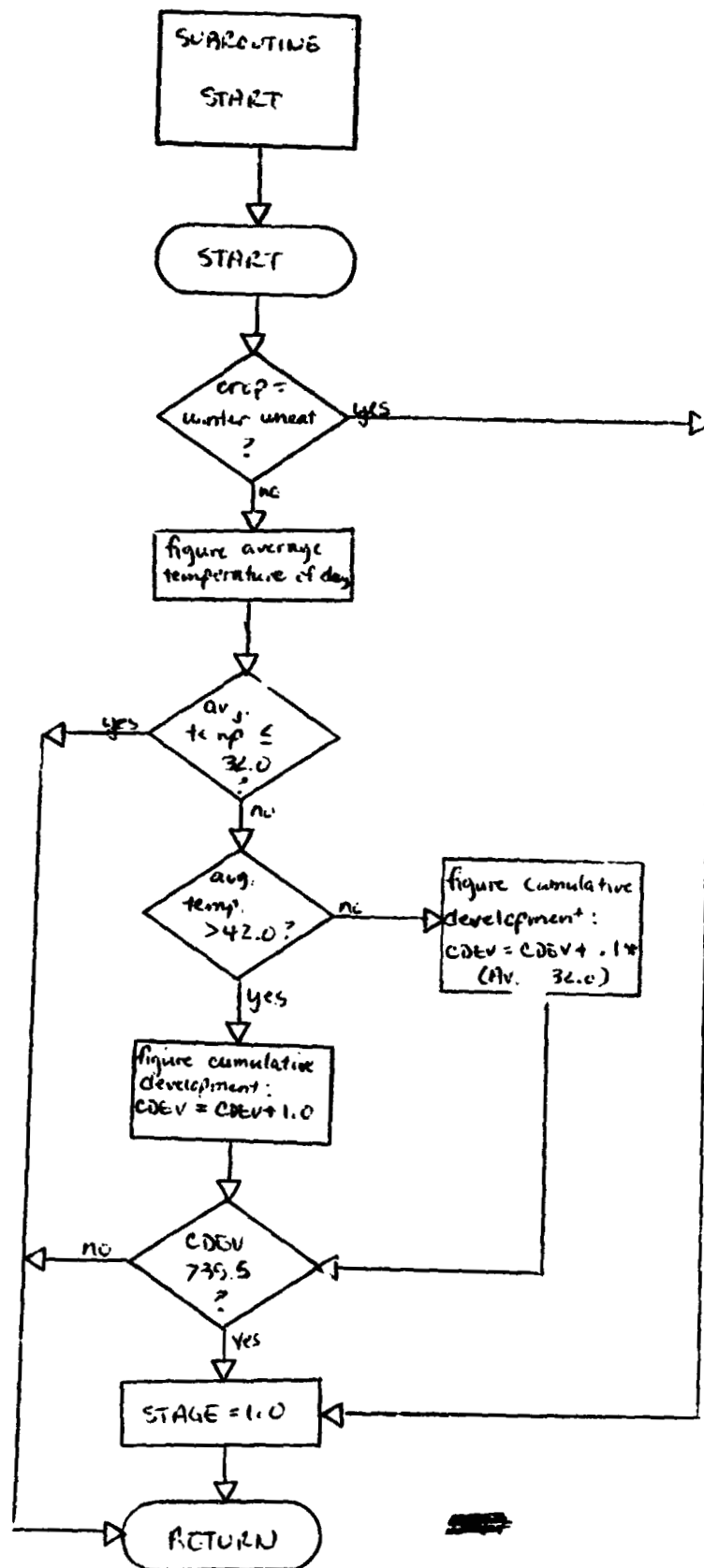
Subroutine START automatically sets STAGE to 1.0 for winter wheat stations. Spring wheat stations must reach a CDEV>35.5 before STAGE is set to 1.0. For average daily temperatures greater than 42°F, CDEV is increased by the value of 1.0. When average daily temperatures fall between 32°F and 42°F, CDEV is increased by one tenth the difference between the day's average and 32°F.

#### 3.2.18.5 Flowchart

Next page.

#### 3.2.18.6 Listing

Follows flowchart.







### 3.2.19 SUBROUTINE PHENO

PHENO is the routine which converts max and min temperatures and daylength into crop progress via Robertson's model with Feyerherm's modifications.

#### 3.2.19.1 Linkages

PHENO calls real function DL.

#### 3.2.19.2 Interfaces

None.

#### 3.2.19.3 Inputs and Outputs

TODAY	Julian day being considered
STAGE	value of Robertson Scale
CROP	winter wheat = 11, spring wheat = 12
MIN	minimum temperature for a given day
MAX	maximum temperature for a given day
MULT	an array of 5 stage- and station-specific multipliers
ALAT	station latitude in hundredths of degrees
ALONG	station longitude in hundredths of degrees

#### 3.2.19.4 Description

Subroutine PHENO calls function DL to compute the daylength of the day in question. The daylength, maximum and minimum temperatures are adjusted according to the stage of the previous day. Three stage advance factors are calculated using stage-specific coefficients: V1 uses daylength, V2 maximum temperature, and V3 minimum temperature. AINCR, the day's crop progress, is computed using the three stage advance factors and a station- and stage-specific multiplier; this is added to STAGE. If the crop stage is advancing too quickly during the fall, STAGE is held constant until a certain Julian date is reached. A check is made to see if the integer portion of STAGE has changed during the above calculations. If no change has occurred, operations return to the main program. When change in the integer portion of STAGE does occur, the amount of change (VT) is calculated, and the day's progress

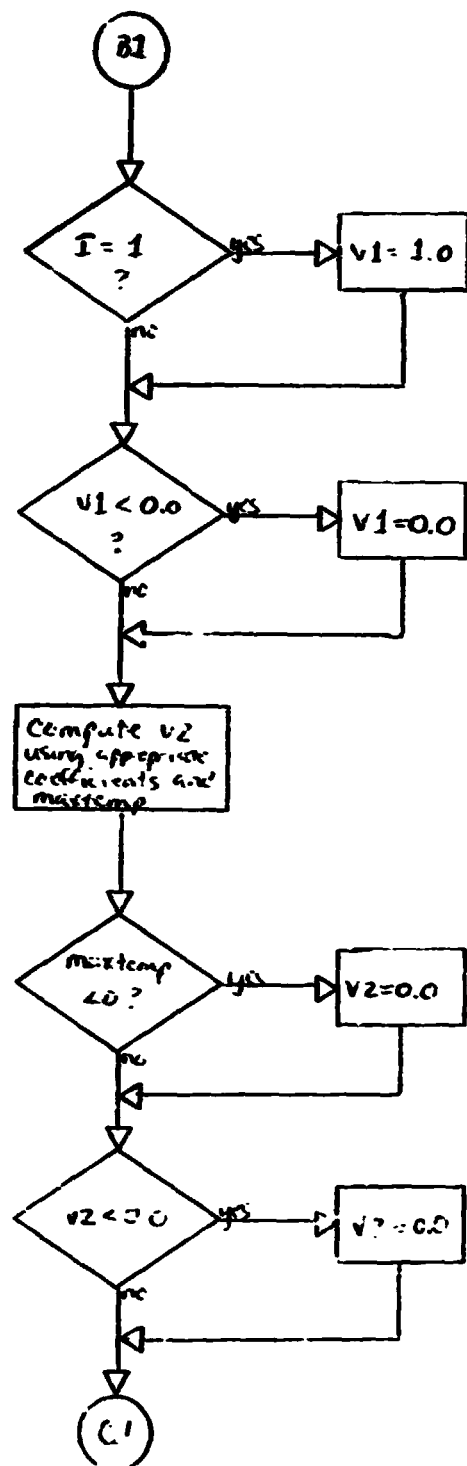
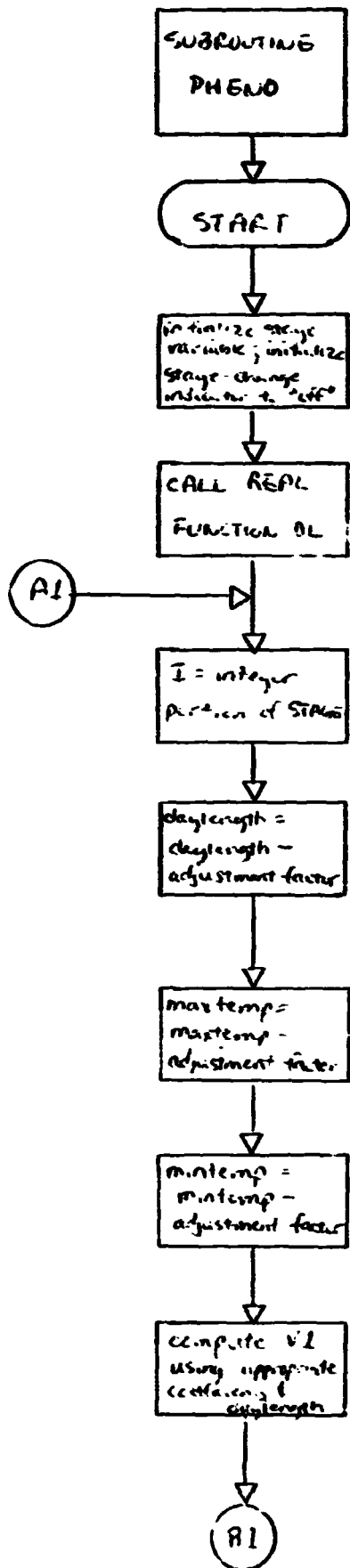
is refigured. The value of STAGE to be returned to the main program is then calculated, adding the integer portion of STAGE to the product of the PCNT and AINCR.

#### 3.2.19.5 Flowchart

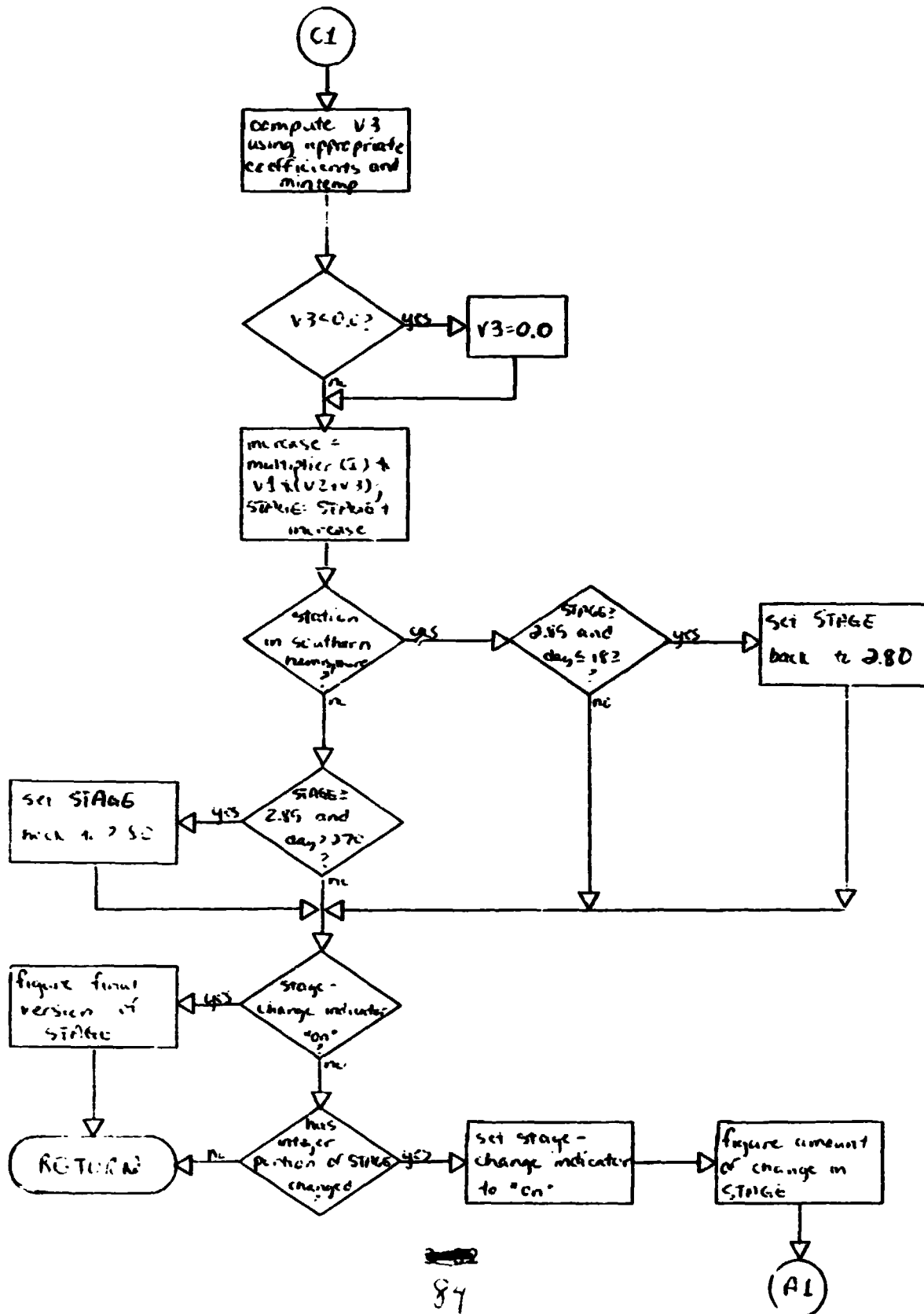
Next page.

#### 3.2.19.6 Listing

Follows flowchart.



# SUBROUTINE PHEND, CONT



```

SUBROUTINE PHENO(TODAY,STAGE,CROP,MIN,MAX,MULT,ALAT,ALONG)
  INTEGER TODAY,CROP,MIN,MAX
  REAL STAGE,ALAT,ALONG,MULT(5),C(5,5),I,IN,AINCR,V1,V2,V3
  DATA C(1,1)/.577/.C(1,2)/.0155/.C(1,3)/-.000223/
  DATA C(1,4)/.00732/.C(1,5)/-.000257/.C(2,1)/4.414/
  DATA C(2,2)/1.005/.C(2,3)/0.07/.C(2,4)/13.64/.C(2,5)/-.003512/
  DATA C(2,6)/.0000525/.C(2,7)/.0003555/.C(2,8)/-.000004282/
  DATA C(3,1)/10.43/.C(3,2)/.055/.C(3,3)/-.00025/.C(3,4)/42.65/
  DATA C(3,5)/.000254/.C(3,6)/0.07/.C(3,7)/.0003943/.C(3,8)/0.0/
  DATA C(4,1)/10.94/.C(4,2)/1.349/.C(4,3)/-.00191/.C(4,4)/42.18/
  DATA C(4,5)/.000254/.C(4,6)/0.07/.C(4,7)/.00003109/.C(4,8)/0.0/
  DATA C(5,1)/24.34/.C(5,2)/-1.14/.C(5,3)/0.0/.C(5,4)/37.67/
  DATA C(5,5)/.0006733/.C(5,6)/0.0/.C(5,7)/.0003442/.C(5,8)/0.0/
  OLDSTG=STAGE
  I=0

```

```

  H=OL (ALAT,TODAY)
  TX=MAX
  TN=MIN

```

10 I=STAGE

```

  H=H-C(1,1)
  TX=TX-C(1,4)
  TN=TN-C(1,4)
  V1=C(1,2)*H+C(1,3)*H*H
  IF (V1.LT.0.0) V1=0.0
  IF (V1.LT.0.0) V1=0.0
  V2=C(1,5)*TX+C(1,6)*TX*TX
  IF (V2.LT.0.0) V2=0.0
  IF (V2.LT.0.0) V2=0.0
  V3=C(1,7)*TN+C(1,8)*TN*TN
  IF (V3.LT.0.0) V3=0.0
  AINCR=MULT(1)+V1*(V2+V3)

```

```

  STAGE=STAGE+AINCR
  IF (ALAT.LT.0.0) GO TO 1
  IF (STAGE.GE.2.45.AND.TODAY.GT.270) STAGE=2.40
  GO TO 2

```

2 1 IF (STAGE.GE.2.45.AND.TODAY.LE.182) STAGE=2.40

```

  IF (INT(OLDSTG) .NE. INT(STAGE)) GO TO 11
  N=INT(OLDSTG)
  N=INT(STAGE)
  IF (N.EQ.0) GO TO 3
  IOT=1
  PCNT=(1.0-(N-OLDSTG)/AINCR)
  GO TO 10

```

3 11 STAGE=N+PCNT\*AINCR

```

  SETUP
  END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

### 3.2.20 FUNCTION DL

DL converts latitude and date to daylength in hours.

#### 3.2.20.1 Linkage

None.

#### 3.2.20.2 Interfaces

None.

#### 3.2.20.3 Inputs

XLAT	station latitude in hundreths of degrees
DATE	Julian day being considered

#### 3.2.20.4 Outputs

Daylength in hours.

#### 3.2.20.5 Description

After initializing constants, the function calculates the angle of the sun on the equator for the date and then adjusts the angle according to the station's latitude. The daylength is computed using the product of the arc cosine of this angle and a constant value.

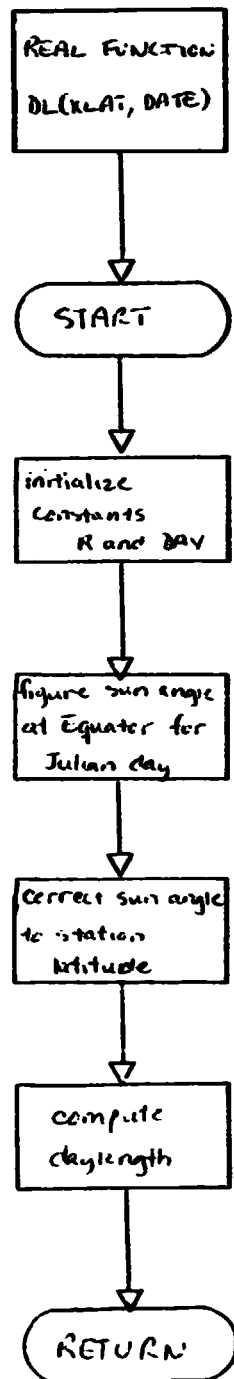
#### 3.2.20.6 Flowchart

Next page.

#### 3.2.20.7 Listing

Follows flowchart.

~~3-84~~



~~3-25~~

```

REAL FUNCTION DI (XLAT,DATE)
REAL XLAT,DAY,EPH,COM
INTEGER DATE
Z=0.017532525
DAY = DATE
EPH=23.5* SIN(0.99-3*(DAY-80.0)*PI)
COM = -TAN(XLAT*PI)*TAN(EPH*PI)
DI=ARCCOS(COM)*7.6408787
RETURN
END

```

ORIGINAL PAGE IS  
OF POOR QUALITY



### 3.2.21 PROGRAM INSDEL

INSDEL is the main routine of the data replacement module. It allows the individual replacement of data in the weather file.

#### 3.2.21.1 Linkages

Program INSDEL call subroutines INIT, SWAP and TERM.

#### 3.2.21.2 Interfaces

INSDEL is run only when the EDIT procedure cannot satisfactorily edit the entire contents of CROPDATA. If needed, INSDEL is executed before program JSCTAPE can be run.

#### 3.2.21.3 Inputs

The INDEX file is on Unit 2, CROPDATA on Unit 4, and the card input containing data corrections is on Unit 5.

#### 3.2.21.4 Outputs

Corrections are made in CROPDATA. Some error messages may appear.

#### 3.2.21.5 Description

After calling INIT, INSDEL reads a card and scans it to see if it is an 'INS'ert or a 'DEL'ete card. Using SWAP, INSDEL then reads the span of days and type of temperature to be changed. If 'DEL' was indicated, the appropriate temperatures are set to missing (9999); if INS was called for, a card containing the new temperatures is read and then temperatures are inserted into CROPDATA. Any time problems are encountered with the card input, an error message is printed and the program proceeds to the next card. After all cards have been processed, TERM is called.

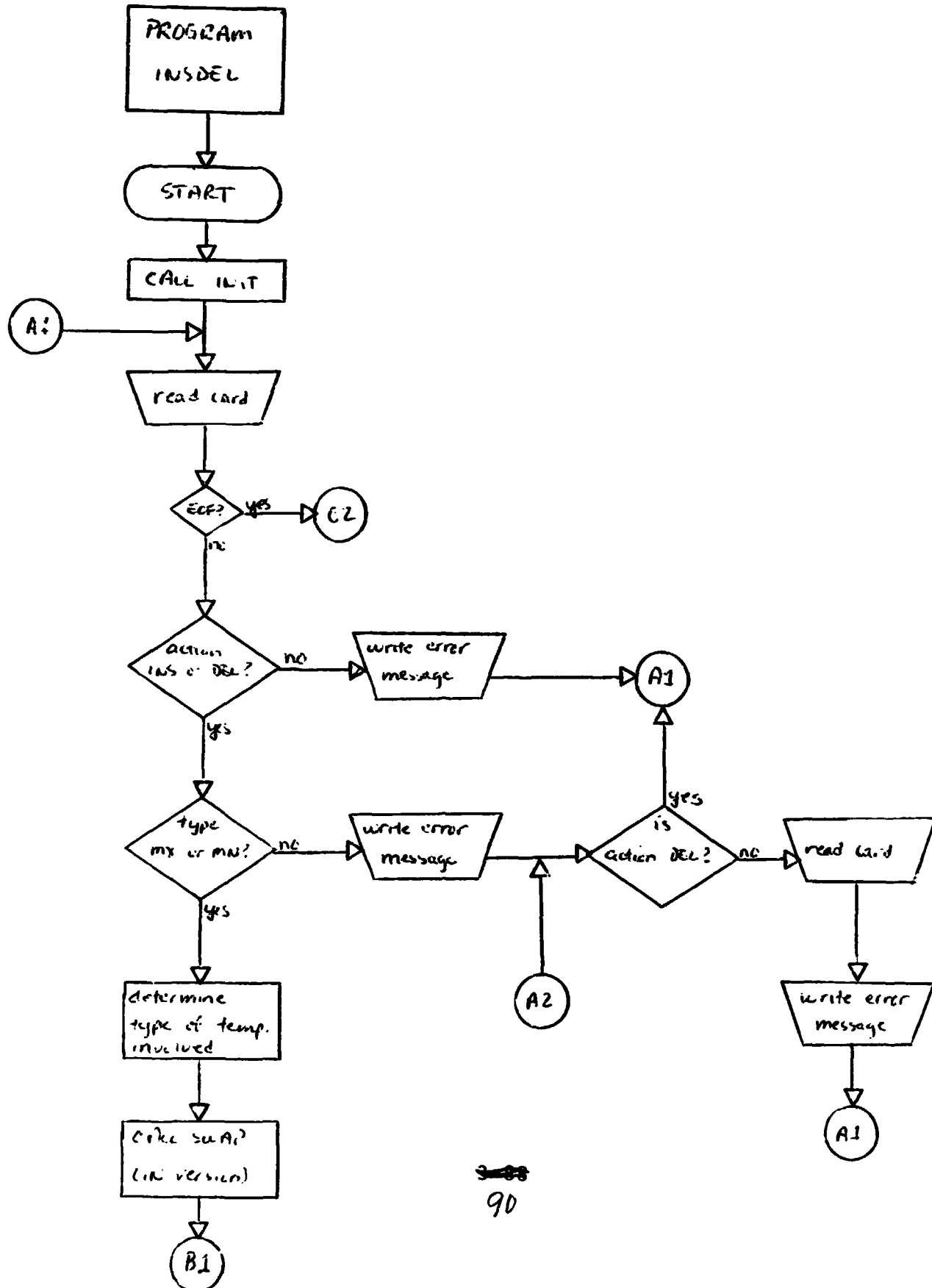
#### 3.2.21.6 Flowchart

Next page.

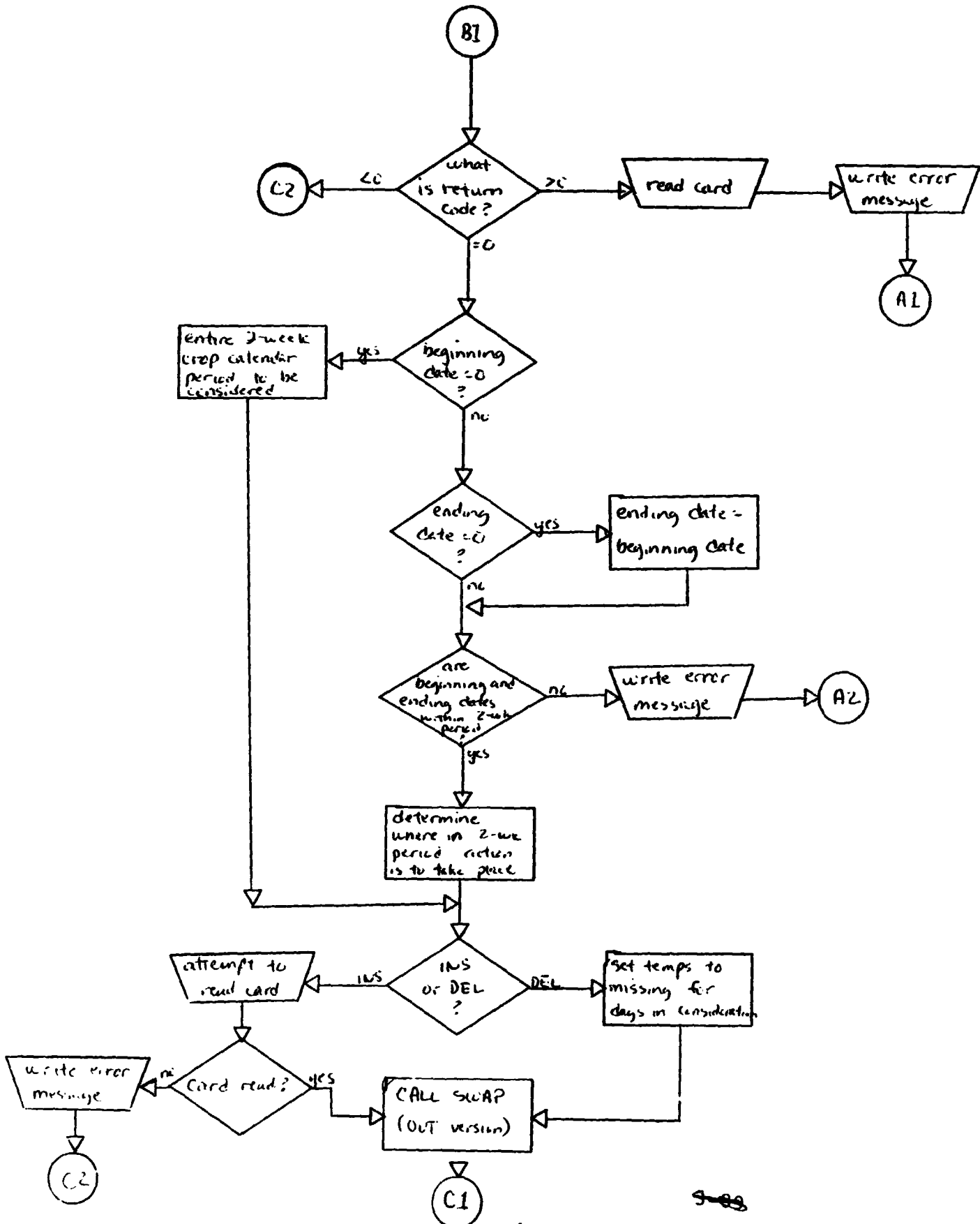
#### 3.2.21.7 Listing

Follows flowchart.

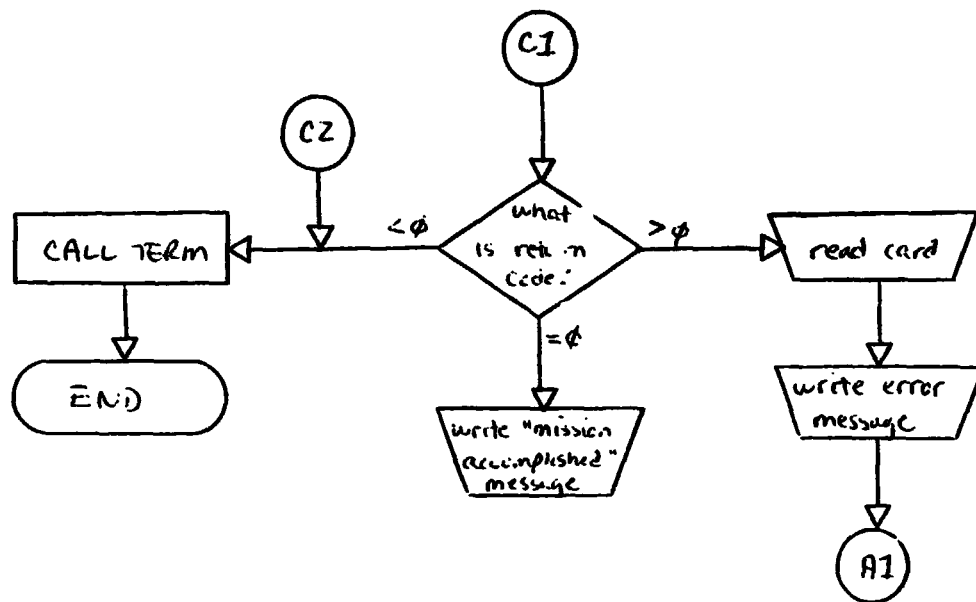
~~9-89~~



# PROGRAM INSDCL, CON'T



PROGRAM INSDEL, CONT



~~3-3-3~~



100 5.1 105 COMMAND IGNORE 1.1)  
DEXTRO 3  
CALL 15-4  
STOP  
END

ORIGINAL PAGE IS  
OF POOR QUALITY

U  
-2

94

### 3.2.22 JCL FILE INSERT

Executing JCL file INSERT invokes the INSDEL program.

#### 3.2.22.1 Linkages

None.

#### 3.2.22.2 Interfaces

None.

#### 3.2.22.3 Inputs

Cards containing the information necessary to correct CROPDATA must be inserted. See Section 4.2.2.5 for formats.

#### 3.2.22.4 Outputs

None.

#### 3.2.22.5 Description

JCL file INSERT binds the files referenced by INSDEL to that program and requests the execution of INSDEL.

#### 3.2.22.6 Listing

```
//INSERT PROC  
//INS EXEC PGM=INSDEL  
//STEPLIB DD DSN=W.EDS.CCEA.PHASE3.LOAD,DISP=SHR  
//FT02F001 DD DSN=W.EDS.CCEA.{GROUP I } INDEX,DISP=SHR  
//FT04F001 DD DSN=W.EDS.CCEA.{GROUP I } CROPDATA,DISP=SHR  
//FT05F001 DD DSN=SYSIN  
//FT06F001 DD SYSOUT=A  
//****END
```

### 3.2.23 PROGRAM DRECOVER

This program fills the INDEX file from card input.

#### 3.2.23.1 Linkages

None.

#### 3.2.23.2 Interfaces

Whenever changes are to be made in the INDEX file, DRECOVER is run to reconstruct the entire file.

#### 3.2.23.3 Inputs

Input to DRECOVER is by cards; the first card contains the number of stations to be in the INDEX file, punched in columns 1-3. Each subsequent card contains a station to be in the file, and its three closest neighbor stations, in 4(I5,1X) format.

#### 3.2.23.4 Outputs

The INDEX file is created.

#### 3.2.23.5 Description

DRECOVER reads all input cards, sorting the data into NUMSTA (number of stations) and two arrays: WMO, containing all stations on the INDEX file, and STAIN, containing the three closest stations to each station in array, WMO. DRECOVER then writes all data to the INDEX file.

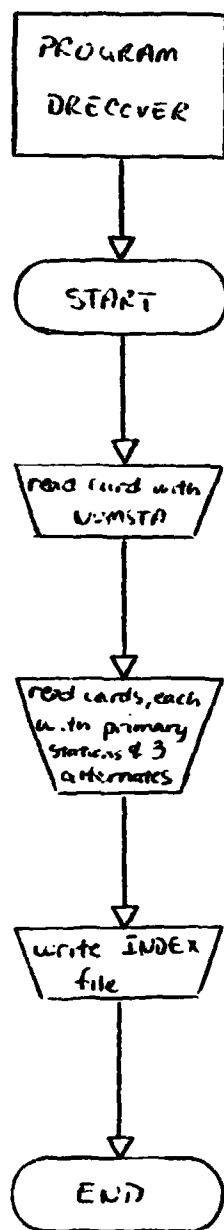
#### 3.2.23.6 Flowchart

Next page.

#### 3.2.23.7 Listing

Follows flowchart.





//DRECOVER JOB ('DD10008E1HEA ', 'COLUM'), VONHOLT, REGION=256K, TIME=1

//SF EXEC NCDGOCGO

//SYSIN DD \*

INTEGER WMO(511), STATN(511,3)

INTEGER\*2 NUMSTA

READ(5,1,END=2) NUMSTA, (WMO(I), (STATN(I,J), J=1,3), I=1,511)

1 FORMAT(13,/,255(4(15,1X),/),255(4(15,1X),/))

2 WRITE(8,3) NUMSTA, (WMO(I), (STATN(I,J), J=1,3), I=1,511)

3 FORMAT(A2,2X,255A4,/,10(2(128A4),/))

STOP

END

//SYSGO DD \*

96

### 3.2.24 PROGRAM JSCTAPE

JSCTAPE is a PL/I program transferring data from disk file GRIDINFO to a tape to be sent to Houston.

#### 3.2.24.1 Linkages

None.

#### 3.2.24.2 Interfaces

JSCTAPE should not be executed until an error-free run of CROPCALN has been processed.

#### 3.2.24.3 Inputs

Disk file GRIDINFO and a card containing header information for the tape. See Section 4.2.3 for card format.

#### 3.2.24.4 Outputs

JSCTAPE outputs a tape copy of file GRIDINFO preceded by a header indicating the countries and Julian dates involved.

#### 3.2.24.5 Description

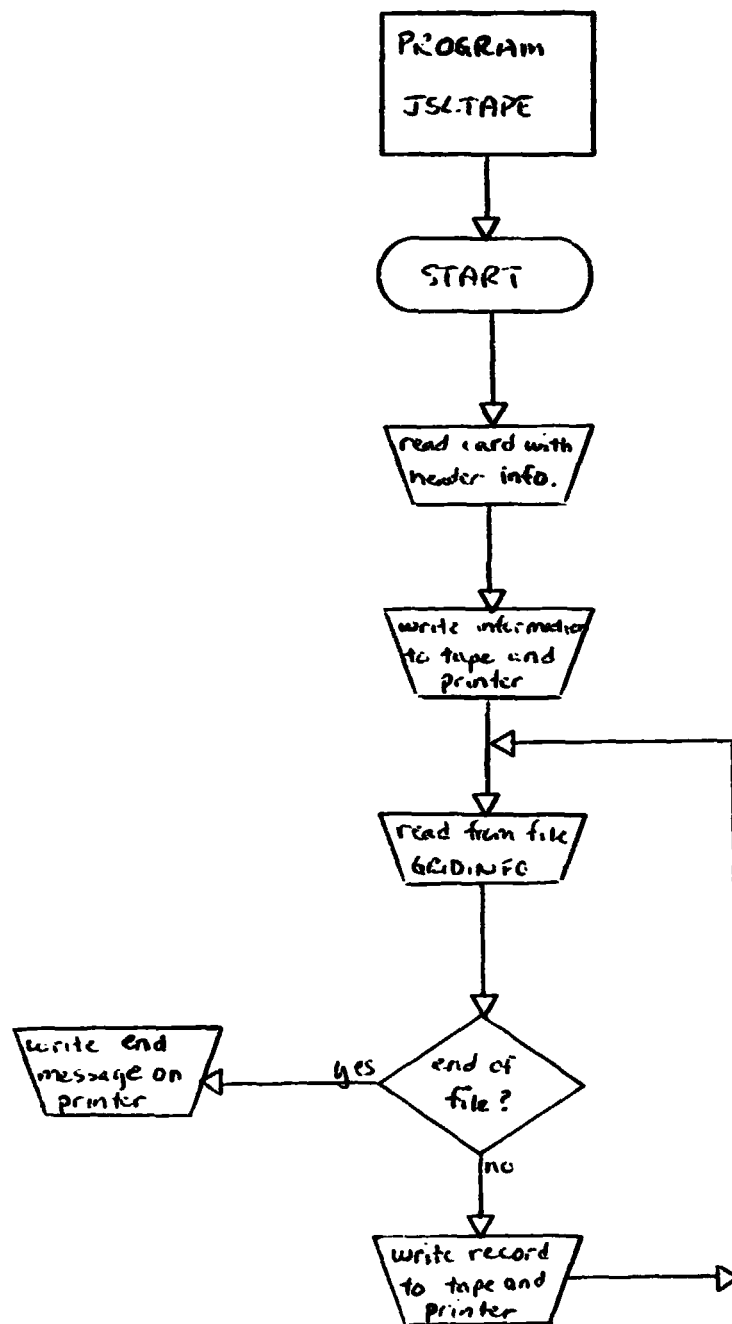
The program reads the card with header information and copies this information both to tape and to the printer. JSCTAPE proceeds to read each record of GRIDINFO and write the data to both the tape and the printer. When end-of-file is encountered, a closing message is written to the printer and operation ceases.

#### 3.2.24.6 Flowchart

Next page.

#### 3.2.24.7 Listing

Follows flowchart.



```

JCTAPE: PROC OPTIONS(MAIN):
DOU INSTR CH$P(80);
DOU SYSIN FILE STREAM INPUT;
DOU SYSPRINT FILE PRINT OUTPUT;
DOU IN FILE RECORD INPUT;
DOU OUT FILE RECORD OUTPUT;
ON ENDFILE(IN) GOTO EXIT;
OPEN FILE(SYSIN). FILE(SYSPRINT). FILE(IN). FILE(OUT);
INSTR=1;
GET FILE(SYSIN) FORT(INSTR)(COL(1).A(80));
WRITE FILE(OUT) FORT(INSTR);
PUT FILE(SYSPRINT) FORT(INSTR)(SKIP.A(80));
1100: READ FILE(IN) INTO(INSTR);
WRITE FILE(OUT) FORT(INSTR);
PUT FILE(SYSPRINT) FORT(INSTR)(SKIP.A(80));
GOTO 1100;
EXIT: PUT FILE(SYSPRINT) FORT(1000 END OF FILE IN 000)(SKIP(3).A);
CLOSE FILE(SYSIN). FILE(SYSPRINT). FILE(IN). FILE(OUT);
END JCTAPE:

```

101

ORIGINAL PAGE IS  
OF POOR QUALITY

## 4. OPERATION

Operation of the Robertson Phenological Model is a mixed human/machine procedure. This document describes the mechanical procedures.

### 4.1 USER DOCUMENTATION

Most of the work involved in operating the crop calendar is in the preparation of meteorological (met) data. To do this properly requires a knowledge of the organization of the crop calendar and met data files.

Crop calendar countries are separated into two sets: Group I - the U.S., China, and India, and Group II - Canada and the U.S.S.R. Each set is run every two weeks with any given 14-day period ending on a Sunday, and the groups are run in alternating weeks (Group I one week, Group II the next). Therefore the 14-day periods comprising each crop calendar run for each group are not identical. For example, the 2-week periods in Julian days for Group I may be 52-65, 66-79, and 80-93, while the 14-day periods for Group II may be 59-72, 73-86, and 87-100.

Met data input is divided by months and by area (North America and non-North America). Previous months' data are on tape, while the current month's data are on disk. For each group there is a maximum of four met data collection runs per two-week period. (last month, this month) x (North America, non-North America).

### 4.2 PROCEDURES

In total, eight procedures are involved: one to construct the INDEX file, six to prepare data and run the Robertson Phenological Model, and one to prepare a tape of crop calendar data for use in Houston.

#### 4.2.1 CONSTRUCTING THE INDEX FILE (DRECOVER)

Whenever changes are to be made in the INDEX file, DRECOVER is run to reconstruct the entire file. Card input follows the program cards as in the example below:

```
//FT08F001DDDSN=W.EDS.CCEA.{GROUP I}.INDEX,DISP=SHR
                                {GROUP II}
//SYSGODD*
235
42809DD42971DD42798DD42492
43003DD43192DD43014DD43117
etc.
/*
//
```

#### 4.2.2 CROP CALENDAR PROCEDURES

There are two sets of Job Control Language (JCL) PROC's; one for the Group I countries and one for Group II. The JCL accesses the partitioned dataset W.EDS.CCEA.PHASE3.LOAD containing all crop calendar programs in executable form. The deck for the appropriate group is input each time any crop calendar program is run. The EXEC card (or cards) for each program in the following sections of explanation refers to a PROC contained in these JCL decks, and is placed after the final ~~DD~~ PEND card in the JCL deck.

The JCL decks for Group I and Group II are found on the following two pages.

# JCL PROCEDURES FOR

## GROUP I

```

//LIST PROC
//LIST EXEC PGM=PRINT
//STEPLIB DD DSN=W.EDS.CCEA.PHASE3.LOAD.DISP=SHR
//FT02F001 DD DSN=W.EDS.CCEA.GROUP1.INDEX.DISP=SHR
//FT04F001 DD DSN=W.EDS.CCEA.GROUP1.CROPDATA.DISP=SHR
//FT06F001 DD SYSOUT=A
//      PEND
//EDIT PROC
//EDIT EXEC PGM=EDITOR,TIME=(,15)
//STEPLIB DD DSN=W.EDS.CCEA.PHASE3.LOAD.DISP=SHR
//FT02F001 DD DSN=W.EDS.CCEA.GROUP1.INDEX.DISP=SHR
//FT04F001 DD DSN=W.EDS.CCEA.GROUP1.CROPDATA.DISP=SHR
//FT05F001 DD DUMMY
//FT06F001 DD SYSOUT=A
//      PEND
//SUZYQ PROC OLDE=,NEW=,DSP=MOD
//TRACK EXEC PGM=CROPCALN
//STEPLIB DD DSN=W.EDS.CCEA.PHASE3.LOAD.DISP=SHR
//FT01F001 DD DSN=W.EDS.CCEA.GROUP1.MASTER&OLD.,DISP=SHR,DCB=BUFNO=1
//FT02F001 DD DSN=W.EDS.CCEA.GROUP1.INDEX.DISP=SHR,DCB=BUFNO=1
//FT04F001 DD DSN=W.EDS.CCEA.GROUP1.CROPDATA.DISP=SHR,DCB=BUFNO=1
//FT06F001 DD SYSOUT=A
//FT08F001 DD DSN=W.EDS.CCEA.GROUP1.GRIDINFO.DISP=&DSP.,DCB=BUFNO=1
//FT09F001 DD DSN=W.EDS.CCEA.GROUP1.MASTER&NEW.,DISP=SHR,DCB=BUFNO=1
//FT11F001 DD DSN=W.EDS.CCEA.GROUP1.ARNO.DISP=SHR,DCB=BUFNO=1
//REPORT EXEC PGM=IESPTCH
//SYSUT1 DD DSN=W.EDS.CCEA.GROUP1.ARNO.DISP=SHR
//SYSIN DD DSN=W.EDS.CCEA.DATALIB(PREFORM).DISP=SHR
//SYSPRINT DD DUMMY
//SYSUT2 DD SYSOUT=A
//      PEND
//CLEAR PROC
//CLEAR EXEC PGM=CLEAR
//STEPLIB DD DSN=W.EDS.CCEA.PHASE3.LOAD.DISP=SHR
//FT02F001 DD DSN=W.EDS.CCEA.GROUP1.INDEX.DISP=SHR
//FT04F001 DD DSN=W.EDS.CCEA.GROUP1.CROPDATA.DISP=SHR
//FT06F001 DD SYSOUT=A
//FT05F001 DD DDNAME=SYSIN
//      PEND
//INSERT PROC
//INS EXEC PGM=INSDI
//STEPLIB DD DSN=W.EDS.CCEA.PHASE3.LOAD.DISP=SHR
//FT02F001 DD DSN=W.EDS.CCEA.GROUP1.INDEX.DISP=SHR
//FT04F001 DD DSN=W.EDS.CCEA.GROUP1.CROPDATA.DISP=SHR
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//      PEND
//CAPTURE PROC UNIT=,VOL=,DSN=W.EDS.CCEA.DAYDATA
//CAPTURE EXEC PGM=CAPTURE
//STEPLIB DD DSN=W.EDS.CCEA.PHASE3.LOAD.DISP=SHR
//FT01F001 DD SYSOUT=A,DCB=(LRECL=100,BLKSIZE=1500,RECFM=FB)
//FT02F001 DD DSN=W.EDS.CCEA.GROUP1.INDEX.DISP=SHR
//FT03F001 DD DSN=&DSN.,DISP=SHR,UNIT=&UNIT.,VOL=SER=&VOL,LABEL=(,.,IN)
//FT04F001 DD DSN=W.EDS.CCEA.GROUP1.CROPDATA.DISP=SHR
//FT06F001 DD SYSOUT=A
//      PEND

```

ORIGINAL PAGE IS  
OF POOR QUALITY



# JCL PROCEDURES FOR

## GROUP II

```

//LIST PROC
//LIST EXEC PGM=PRINT
//STEPLIB DD DSN=W.EDS.CCEA.PHASE3.LOAD,DISP=SHR
//FT02F001 DD DSN=W.EDS.CCEA.GROUP11.INDEX,DISP=SHR
//FT04F001 DD DSN=W.EDS.CCEA.GROUP11.CROPDATA,DISP=SHR
//FT06F001 DD SYSOUT=A
//      PEND
//EDIT PROC
//EDIT EXEC PGM=EDITOR,TIME=(,15)
//STEPLIB DD DSN=W.EDS.CCEA.PHASE3.LOAD,DISP=SHR
//FT02F001 DD DSN=W.EDS.CCEA.GROUP11.INDEX,DISP=SHR
//FT04F001 DD DSN=W.EDS.CCEA.GROUP11.CROPDATA,DISP=SHR
//FT06F001 DD SYSOUT=A
//      PEND
//SUZYD PROC OLD=,NEW=,DSP=MOD
//TPACK EXEC PGM=CROPCALN
//STEPLIB DD DSN=W.EDS.CCEA.PHASE3.LOAD,DISP=SHR
//FT01F001 DD DSN=W.EDS.CCEA.GROUP11.MASTER&OLD.,DISP=SHR,DCB=BUFNO=1
//FT02F001 DD DSN=W.EDS.CCEA.GROUP11.INDEX,DISP=SHR,DCB=BUFNO=1
//FT04F001 DD DSN=W.EDS.CCEA.GROUP11.CROPDATA,DISP=SHR,DCB=BUFNO=1
//FT06F001 DD SYSOUT=A
//FT08F001 DD DSN=W.EDS.CCEA.GROUP11.GRIDINFO,DISP=SHR,DCB=BUFNO=1
//FT09F001 DD DSN=W.EDS.CCEA.GROUP11.MASTER&NEW.,DISP=SHR,DCB=BUFNO=1
//FT11F001 DD DSN=W.EDS.CCEA.GROUP11.ARNO,DISP=SHR,DCB=BUFNO=1
//REPORT EXEC PGM=IEBPTPCH
//SYSUT1 DD DSN=W.EDS.CCEA.GROUP11.APNO,DISP=SHR
//SYSIN DD DSN=W.EDS.CCEA.DATALIB(PREFORM),DISP=SHR
//SYSPRINT DD DUMMY
//SYSUT2 DD SYSOUT=A
//      PEND
//INSEPT PROC
//INS EXEC PGM=INSEDI
//STEPLIB DD DSN=W.EDS.CCEA.PHASE3.LOAD,DISP=SHR
//FT02F001 DD DSN=W.EDS.CCEA.GROUP11.INDEX,DISP=SHR
//FT04F001 DD DSN=W.EDS.CCEA.GROUP11.CROPDATA,DISP=SHR
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//      PEND
//CLEAR PROC
//CLEAR EXEC PGM=CLEAR
//STEPLIB DD DSN=W.EDS.CCEA.PHASE3.LOAD,DISP=SHR
//FT02F001 DD DSN=W.EDS.CCEA.GROUP11.INDEX,DISP=SHR
//FT04F001 DD DSN=W.EDS.CCEA.GROUP11.CROPDATA,DISP=SHR
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//      PEND
//CAPTURE PROC UNIT=,VOL=,DSN='W.EDS.CCEA.DAYDATA'
//CAPTURE EXEC PGM=CAPTURE
//STEPLIB DD DSN=W.EDS.CCEA.PHASE3.LOAD,DISP=SHR
//FT01F001 DD SYSOUT=A,DCB=(LRECL=100,BLKSI7E=1500,RECFM=FB)
//FT02F001 DD DSN=W.EDS.CCEA.GROUP11.INDEX,DISP=SHR
//FT03F001 DD DSN=,DISP=SHR,UNIT=,VOL=SER=,LABEL=(,IN
//FT04F001 DD DSN=W.EDS.CCEA.GROUP11.CROPDATA,DISP=SHR
//FT06F001 DD SYSOUT=A
//      PEND

```

#### 4.2.2.1 INITIALIZING FILE CROPDATA (CLEAR)

The CLEAR program is called by the CLEAR procedure. This sets all temperatures in W.EDS.CCEA. $\left\{ \begin{array}{l} \text{GROUP I} \\ \text{GROUP II} \end{array} \right\}$ .CROPDATA to '9999'. The starting date for the crop calendar period being run is also determined by CLEAR. The default option will add 14 to the starting date of the last period to get the new starting date for the period being run. This may be overridden by user input, which is required if the previous two-week period spans two years. Examples are given below:

- A. The default option.

```
//EXEC CLEAR
```

- B. Setting the starting date 28 days after the last period.

```
//EXEC CLEAR
//FT05F001DD*
28
/*
```

- C. Setting the starting date for a new year (in this example, the previous starting date is 362 and the next one is 010; the change is a -352 and the following input is used).

```
//EXEC CLEAR
//FT05F001DD*
-352
/*
```

#### 4.2.2.2 FILLING FILE CROPDATA (CAPTURE)

The CAPTURE program is called by the CAPTURE procedure. This obtains maximum (MX) and minimum (MN) temperature data from catalogued disk files in the current month and from tape in past months. Four datasets can be accessed by CAPTURE:

1. W.EDS.CCEA.DAILY - to access North American data in the current month (disk file).
2. CCEA.xxx.SRTD - to access North American data in a past month (tape access) where 'xxx' is the month abbreviation (ex., CCEA.JAN.SRTD).

~~4.2.2.2~~

3. W.EDS.CCEA.DAYDATA - for non-North American data in the current month (disk file).
4. CCEA.DAILY.xxx - for non-North American data in a past month (tape access), where 'xxx' is the month abbreviation (ex., CCEA.DAILY.JAN.).

Data for past months is put on tape at Suitland, Maryland. The appropriate dataset name (and unit and tape number, if needed) is input on the EXEC card. When accessing North American data (W.EDS.CCEA.DAILY or CCEA.xxx.SRTD) the Julian date of the first day of the month that is being run must be input. This is not needed for non-North American data (W.EDS.CCEA.DAYDATA or CCEA.DAILY.xxx). See examples below:

#### A. GROUP I (China, India, and the U.S.)

To CAPTURE data entirely within the current month, the following input is used:

```
//BEXECCAPTURE,DSN='W.EDS.CCEA.DAYDATA'
//BEXECCAPTURE,DSN='W.EDS.CCEA.DAILY'
//FT05F001BDD*
001      (eg., for January)
/*
```

To CAPTURE data which spans a two-month period, the following input is used:

```
//BEXECCAPTURE,DSN='CCEA.JAN.SRTD',UNIT=TAPE9,VOL=Exxxxx
//FT05F001BDD*
001
/*
//BEXECCAPTURE,DSN='W.EDS.CCEA.DAILY'
//FT05F001BDD*
032
/*
//BEXECCAPTURE,DSN='CCEA.DAILY.JAN',UNIT=TAPE9,VOL=Exxxxx
//BEXECCAPTURE,DSN='W.EDS.CCEA.DAYDATA'
/*
```

#### B. GROUP II (U.S.S.R. and Canada)

When running Group II the CAPTURE program that accesses W.EDS.CCEA.DAYDATA is on the CAPTURE procedure card in the JCL set-up; therefore the EXEC card need only read:  
//BEXECCPGM=CAPTURE. See examples below.

To CAPTURE data entirely within the current month, the following input is used:

```
//EXECPGM=CAPTURE
//EXECCAPTURE,DSN='W.EDS.CCEA.DAILY'
//FT05F001DD*
001    (eg., for January)
/*
```

To CAPTURE data which spans a two-month period, the following input is used:

```
//EXECCAPTURE,DSN='CCEA.JAN.SRTD',UNIT=TAPE9,VOL=Exxxxx
//FT05F001DD*
001    (eg., for January as a past month)
/*
//EXECCAPTURE,DSN='W.EDS.CCEA.DAILY'
//FT05F001DD*
032    (for February)
/*
//EXECCAPTURE,DSN='CCEA.DAILY.JAN',UNIT=TAPE9,VOL=Exxxxx
//EXECPGM=CAPTURE
/*
```

#### C. Spanning two years

Two runs of the CAPTURE program are needed; one will capture data for the end of the first year and the other will capture data for the beginning of the new year.

When CAPTURE is used on North American datasets for the end of the first year, the Julian date for December 1 is input as it was for previous December runs. Nothing else is required.

When CAPTURE is used on North American datasets for the beginning of the new year, the Julian date for January 1, 001, is input as expected. On the next input card the value of the new year is punched:

```
//EXECCAPTURE,DSN='appropriate name'
//FT05F001DD*
335
/*
//EXECCAPTURE,DSN='appropriate name'
//FT05F001DD*
001
1977
/*
```

When CAPTURE is used on non-North American datasets, no input

is needed on either run because both the year and the Julian date can be read from the dataset.

In all cases when two years are spanned, the year is needed to determine if the first year is a leap year so that the Julian dates greater than 365 can be adjusted correctly.

#### 4.2.2.3 EDITING FILE CROPDATA (EDIT)

The EDITOR program is called by the procedure EDIT, and checks for inverted temperatures or missing data. There are three checks and two edits in this procedure. EDITOR scans the data twice and replaces or corrects missing data either by using data from the three closest stations or statistics generated by the primary station to find a substitute. The third time EDITOR checks the data and prints it out indicating the stations it could not correct, and stations where there are days with an unusually large or small gap between the MX and MN. The input for both GROUP I and GROUP II is as follows:

```
//BEXECBEDIT
//FT05F001BDDB*
BBBBB1BBB1BBB0
BBBBB1BBB1BBB0
BBBBB2BBBBBBB1
/*
```

The EDIT PROC needs 250K.

To correct the data through input, use the INSERT procedure, explained in Section 4.2.2.5

#### 4.2.2.4 RUNNING THE ROBERTSON PHENOLOGICAL MODEL (SUZYQ)

SUZYQ is the Crop Calendar procedure. It executes program CROPCALN which uses the daily temperatures to compute how much the crop should have progressed each day for the 14-day period. SUZYQ prints out to three files: GRIDINFO, which contains the daily increments of the crop stage and is subsequently put onto tape; ARNO, which allows the summary data (the last day of the period) to be accessed from a remote terminal in Houston; and MASTER, which has two files, 1 and 2, that writes the daily increments to the "old" file and the stage of the 14th day to the "new" file. The input is as follows:

```
//BEXECBSUZYQ,OLD=1,NEW=2,DISP=OLD
or //BEXECBSUZYQ,OLD=2,NEW=1,DISP=OLD
```

A record must be kept of which Master file (1 or 2) is used for input and output for each period (i.e., for OLD and NEW). The 14th (or last) day of the period is written onto the "new" file each run, which on the next run becomes the "old" file. The Crop Calendar procedure will begin incrementing at the correct date, using the 14th day of the "old" file to calculate the new starting date. The Master files are reversed each time SUZYQ is run. The SUZYQ procedure produces a summary output which prints out the last day of the period and the crop's stage.

It is easiest and most expedient to run EDIT and SUZYQ together. The EDIT "checker" Subroutine prints MX and MN, the difference between them, and indicates the days containing inverted temperatures or a wide or small gap between MX and MN by an 'XX'. SUZYQ prints the stations that have inverted temperatures and the day(s) on which this occurs (day 1, day 2, etc) in the 14-day period. These error messages appear in summary form as part of the crop calendar output. The input would be:

```
//EXEC EDIT
//FT05F001DD*
000000100010000
000000100010000
000000200000001
/*              (2)   (1)
//EXEC SUZYQ,OLD=1,NEW=2,DISP=OLD
/*
```

When the two-week period spans two years, regardless of country, the value of the first year must be input:

```
              (2)   (1)
//EXEC SUZYQ,OLD=1,NEW=2,DISP=OLD
//FT05F001DD*
1976
/*
```

As in the CAPTURE routine, the year is needed to determine if the first year is a leap year, so Julian dates in January can be adjusted correctly.

#### 4.2.2.5 MANUAL EDITING OF FILE CROPDATA (INSERT)

If corrections need to be made for inverted temperatures or missing data, they are made with the INSERT procedure, which calls the program INSDEL. INSDEL will both insert and delete information; the two options are discussed separately.

~~110~~

A. To insert data.

```
//EXECINSERT
//FT05F001DD*
INSXXXXXXmx0000mn0000
aaaaabbbbccccddddd....
/*
```

where: XXXXX is the station number  
mx,mn indicates whether the maximum or minimum temperature is to be changed  
000 the range of days, by Julian date, to be changed. If only one day is changed, the second field may be omitted.  
aaaaa.. the actual change. in fields of 5 (1415)

An example -- changing the maximum temperature for station 43758 to 61 and 63 on days 79 and 80 respectively:

```
INS43758MX079080
6163
```

If only one day for a station is changed, the input would be:

```
INS43758MX079
61
```

Any number of INSERT change cards may be run at the same time, in groups of two as shown in the above examples.

When the INSERT procedure is run, SUZYQ should be run as a separate step after INSERT to be sure the changes were made and SUZYQ calculated the increments for those stations. One way to make a quick check is to make sure the last day of the period is the same Julian date printed out on the SUZYQ crop calendar summary output. The INSERT program also indicates if the correction was successful by messages routed to the printer.

B. To delete data.

```
//EXECINSERT
//FT05F001DD*
DELXXXXXXmx0000mn0000
/*
```

where: XXXXX is the station number  
mx,mn indicates whether maximum or minimum temperature is to be deleted  
000 the range of days, by Julian date to be deleted. If only one day is deleted, the

~~///~~  
///

second field may be omitted.

Ar. example:

```
DEL437580MN083085
DEL437580MX083085
```

This example would replace the maximum and minimum temperature values on days 83 to 85 with '9999' (indicating missing values).

As with the INS option, any number of delete cards may be run at the same time.

#### 4.2.2.6 LISTING OF FILE CROPDATA (LIST)

The LIST procedure calls program PRINT and can be executed at any time to list the contents of the CROPDATA file. In all the procedures that are run for the Crop Calendar (CLEAR, CAPTURE, EDIT, SUZYQ, INSERT) the LIST procedure produces the listing of the data on CROPDATA and should be run with each program.

The input would be: //EXECBLIST

#### 4.2.3 COPYING FILE GRIDINFO TO TAPE (DSJSCTAPE)

This procedure, using program JSCTAPE, is run to put the crop calendar daily increments from file GRIDINFO onto tape to be sent to Houston. The JCL for DJSTAP1 (for GROUP I) or DJSTAP2 for (GROUP II) follows:

```
job card
//TIME=1,CLASS=C
//EXECBNPLIXCLG
//PLI.SYSINDD*
//GO.INDDDSN=W.EDS.CCEA.{GROUP I}.GRIDINFO,DISP=SHR
                        {GROUP II}
//GO.OUTDDUNIT=TAPE7,DISP=(NEW,KEEP),LABEL=(1,BLP,,OUT),
//DDCB=(RECFM=FB,LRECL=80,BLKSIZE=3200,DEN=2,TRTCH=ET),
//DDVOL=SER=Exxxxx
//GO.SYSINDD*
label card
/*
```

On each run the VOL=SER='tape number' must be changed and the appropriate date for the period must be put on the label card



following the card //GO.SYSIN~~DD~~\*. The format for this card is as follows:

columns 1-6	CCCOU
7-8	year of first day of the run period
9-11	Julian date of first day of run period
13-14	year of last day of run period
15-17	Julian date of last day of run period
19-80	commentary - used to indicate the GROUP and countries on the tape

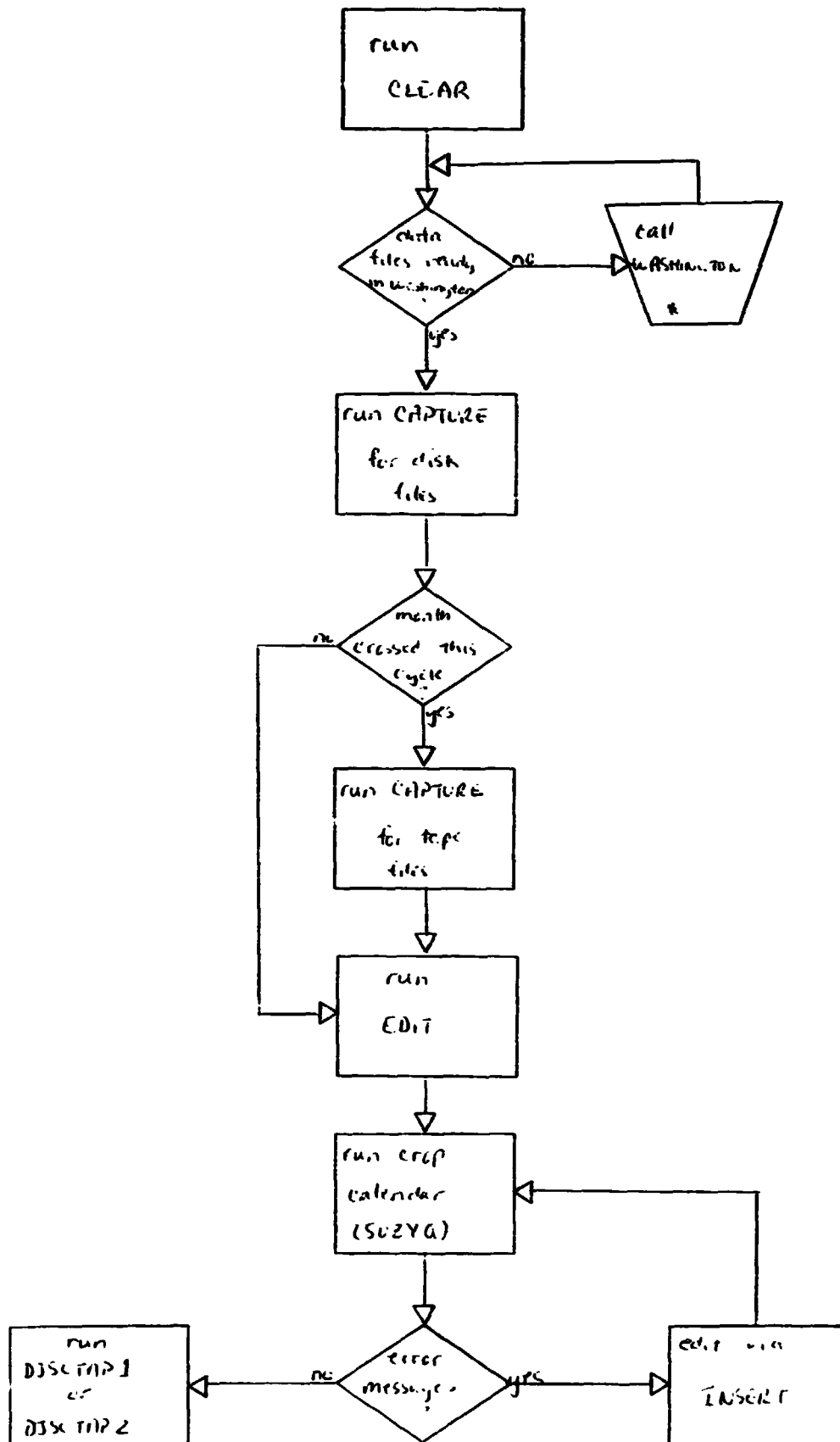
example: CCCOUT77101~~77114~~GROUPII-COUNTRIES:USSR,CANADA\*

After the DJSCTAP job has been successfully run, the tape must be sent to Houston. CCEA/Columbia calls Suitland and gives Mr. Pennington the tape number, the group number, and the Julian dates for that run period. This information is for the outside label.

#### 4.3 FLOWCHART

A flowchart showing the sequence of the crop calendar procedures is found on the following page.

~~113~~



~~114~~